gLOP: A Cleaner Dirty Model for Multitask Learning

by

Rhiannon Rose

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Math
in
Computer Science

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Multitask learning (MTL) was originally defined by Caruana (1997) as "an approach to inductive transfer that improves learning for one task by using the information contained in the training signals of other *related* tasks". In the linear model setting this is often realized as joint feature selection across tasks, where features (but not necessarily coefficient values) are shared across tasks. In later work related to MTL Jalali (2010) observed that sharing all features across all tasks is too restrictive in some cases, as commonly used composite absolute penalties (like the $\ell_{1,\infty}$ norm) encourage not only common feature selection but also common parameter values between settings. Because of this, Jalali proposed an alternative "dirty model" that can leverage shared features even in the case where not all features are shared across settings. The dirty model decomposes the coefficient matrix Θ into a row-sparse matrix B and an elementwise sparse matrix S in order to better capture structural differences between tasks.

Multitask learning problems arise in many contexts, and one of the most pertinent of these is healthcare applications in which we must use data from multiple patients to learn a common predictive model. Often it is impossible to gather enough data from any one patient to accurately train a full predictive model for that patient. Additionally, learning in this context is complicated by the presence of individual differences between patients as well as population-wide effects common to most patients, leading to the need for a dirty model. Two additional challenges for methods applied in the healthcare setting include the need for scalability so that the model can work with big data, and the need for interpretable models. While Jalali gives us a dirty model, this method does not scale as well as many other commonly used methods like the Lasso, and does not have a clean interpretation. This is particularly true in the healthcare domain, as this model does not allow us to interpret coefficients in relation to all settings. Because B coefficients in the dirty model paradigm are not required to be the same for all settings for a particular feature, departures from the global model may be captured in B *or* S leading to ambiguity in interpreting potential main effects.

We propose a "cleaner" dirty model gLOP (global/LOcal Penalty) that is capable of representing global effects between settings as well as local setting-specific effects, much like the ANalysis Of VAriance (ANOVA) test in inferential statistics. However, the goal of the ANOVA is not to build an accurate predictive model, but to identify coefficients that are non-zero at a given level of statistical significance. By combining the dirty model's decomposed Θ matrix and the underlying concept behind the ANOVA, we get the best of both worlds: an interpretable predictive model that can accurately recover the underlying structure of a given problem. gLOP is structured as a coordinate minimization problem

which decomposes Θ into a global vector of coefficients g and a matrix of local setting-specific coefficients L. At each step, g is updated using the standard Lasso paradigm applied to the composite global design matrix in which the design matrices from each setting are concatenated vertically. In contrast, L is updated at each step using the standard Lasso paradigm applied separately to each setting. Another significant advantage of our model gLOP in comparison to previous dirty models is the out-of-the-box use of standard Lasso implementations instead of less frequently implemented CAP family penalties such as the $\ell_{1,\infty}$ norm. Additionally, gLOP has a significant advantage in lowered computational time demands as it takes larger steps towards the global optimum at each iteration. We present experimental results comparing both the runtime and structure recovered by gLOP to Jalali's dirty model.

Acknowledgements

First and foremost, I would like to thank my supervisor Dan Lizotte for being a fantastic mentor and teacher. This thesis would not have been possible without your guidance and contributions, and with your help I have gained an amazing amount of knowledge over the course of this work. I would also like to thank my committee members Robin Cohen and Pascal Poupart for being readers on my thesis and for providing me with valuable feedback on this work. Additionally, I thank Jesse Hoey for all of his support with my transition to the field of Computer Science.

I would like to thank Tejal Patel and Feng Chang for their collaboration and valuable feedback on the use of patient data and clinical research design; you have been an excellent source of knowledge on analgesia and the effects of analgesia administration on patient physiology. I would also like to thank Randall Wetzel at CHLA, and Jesse Ehrenfeld and Warren Sandberg at Vanderbilt University for their guidance on the use of patient data and analgesia administration in the ICU.

In the early days of my undergraduate degree, I was given an invaluable opportunity to participate in research that shaped the course of both my undergraduate and masters degrees, and my future career plans. Eric Roy, thank you for your mentorship, support and guidance. Pascal Poupart, Amanda Clark, David Gonzales and James Tung have also played an extremely important role in my development as a researcher, student and person.

The many friends I have at the University of Waterloo have been an amazing source of support, both emotionally and academically. I would especially like to thank Hella Hoffmann, Valerie Sugarman, Dean Shaft, Cecylia Bocovich, Stephen Kiazyk, Aaron and Christina Moss, Oliver Trujillo, Ben Storer, John Doucette, Dan Recoskie, Sharon Choy and Michael Cormier: your support has been invaluable throughout my adjustment to the field of Computer Science.

My family and close friends outside of the sphere of academia have been a very important source of support and comfort throughout the course of my academic career. My mother Jennifer has been a source of inspiration, guidance, advice and support, and my siblings Samantha, Connor and Maarten and my nephew Calvin are always able to put a smile on my face, even at the most stressful of times. I also really appreciate the support of the Johnstons: Sarah, Annie, Paul and Monique. Since the 6th grade you have welcomed me as part of your family, and I would not be the person I am today without your support. I would like to thank Allie Engelhardt for having an amazing sense of humour and being a great friend. Finally, many thanks to Adam Hartfiel for guiding me through the more challenging (and the less challenging) aspects of computer science and mathematics, and for always having the ability to make me laugh.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis provides a new method of multi-task learning (MTL) for contexts in which features may be shared globally across all settings, but also have setting-specific individual model contributions. Previously, MTL has been used in health-related applications such as severity of illness prediction and triage [7]; in this context, MTL was used to predict mortality of patients with pnuemonia; inputs to the model were a number of demographic and basic physiological measurements such as pulse; and the results of 35 laboratory tests collected after patient admission. However, MTL has not been previously used for predicting analgesia requirements in a postoperative recovery setting, and despite much research in the clinical community on the issue of relevant features for pain (such as facial expression [4],demographic factors [51] and physiological measures such as oxygen levels, heart rate and respiration [2]), and some exploration of features using statistical and machine learning methods (such as regression techniques, machine vision [4, 18], and probabilistic graphical models [53]) few studies propose rigorous quantitative models for pain management using physiological indicators that would provide a prediction of when a patient might require a dose of analgesia in the future using means that do not require subjective assessments of a patient's current level of pain. A detailed review of previous work on features relevant for pain detection, and methods that have been previously used for predicting pain is provided in Appendix A. This work is a first step toward developing a real-time model to predict analgesia demand and to control analgesia administration to improve acute pain management in post operative recovery settings. Because the intended application of this work is in a healthcare context, we desire readily-interpretable predictive models that can accurately represent this underlying structure and are capable of scaling to large data sets.

An extremely simplistic example of such an application would be in the case where we have a patient with several "features" measured at different points in time: for example,

blood pressure (BP), respiration (RESP), oxygen saturation (SPO2) and blood glucose (BG). Our goal is to predict the time (in seconds, minutes or hours) to the next dose of analgesia needed by the patient. Let our example feature vector $x = [\text{BP,RESP,SPO2,BG}]$ where each of these values represents a measurement for the corresponding feature at the same point in time based on current measurements. For the set of measurements observed at this point in time, let our true target be a value y representing how far from the time of this observation of measured features the patient actually required a dose of analgesia. For a randomly generated specific example, our data for one observation is the following:

$$x = \begin{bmatrix} -0.15 & 0.25 & -1.96 & -0.29 \end{bmatrix} y = 0.27.$$

Given a new x and an unknown y, we wish to make predictions of the form:

$$\hat{y} = x\beta$$

where β is the learned vector of weights (i.e. parameters) applied to each feature [22]. In our example, the true set of weights used to generate the data are

$$\beta^* = \begin{bmatrix} 3 & 3 & 0 & 0 \end{bmatrix}.$$

Our goal is to find a parameter vector β that yields good predictions for future instances of data observed from this patient. We accomplish this by gathering many instances or observations of measured features (eg BP, RESP, O2SAT and BG) and corresponding measured targets (time to next dose for each of observation of these features). This is known as our *training data* as it will be used to train the predictive model. This larger set of many instances of the measured features is known as our *design matrix*. In our simplistic example we have five observations of measured features, and five corresponding targets. Our design matrix X and vector of targets y are then

$$X = \begin{bmatrix} -0.15 & 0.25 & -1.96 & -0.29 \\ -1.18 & -1.29 & -0.94 & 0.01 \\ 0.33 & -0.60 & 1.33 & 1.44 \\ 0.95 & 0.14 & -0.33 & -0.53 \\ 0.98 & 0.94 & 0.64 & -0.11 \end{bmatrix} y = \begin{bmatrix} 0.27 \\ -8.16 \\ -0.77 \\ 3.64 \\ 5.24 \end{bmatrix}$$

Specifically, we want to find the vector β that predicts y from our training data X. The quality of a prediction for a single instance with a known target is defined by (for example)

$$(y - x\beta)^2$$

where $(y - x\beta)$ is the *residual*. It is the leftover distance from our predicted value to the true target values after we make our prediction using x and β. Note that smaller values indicate better performance. Supposing we have big data and want good predictions on average over our training data, we learn our overall vector of weights $\hat{\beta}$ using the following equation:

$$\hat{\beta}_{OLS} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i\beta)^2$$

where n is the number of observations (i.e. instances) in the training set, and $\sum_{i=1}^{n}(y_i - x_i\beta)^2$ is the residual sum of squares (RSS) [22], and measures how well a particular β predicts the y in the training set. Hopefully if the training data is large and representative, β will predict well on new, unseen data. Using the notation for our design matrix X, we can rewrite this as

$$\hat{\beta}_{OLS} = \arg\min_{\beta} \|y - X\beta\|_2^2.$$

This is also known as the "Ordinary Least Squares" (OLS) model [48]. In our very simplistic running example, our OLS β estimates would be:

$$\beta_{OLS} = \begin{bmatrix} 1.08 & 3.39 & -0.14 & -0.64 \end{bmatrix}^{\mathsf{T}}$$

However, suppose we have many features and *believe* that most have no impact on the target variable, and thus are not useful for prediction. We can modify our RSS criterion by adding a "penalty" on β that discourages many large, non-zero entries. Simply, this may be written as

$$\arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

where p is the number of features per instance in the design matrix of observations X. This is *not* OLS as it gives a β that does not fit the training data optimally, but hopefully performs better on unseen data than a model fit perfectly to the training data. This is also known as the Least Absolute Shrinkage and Selection Operator (Lasso) [50]. In this paradigm features that are relevant for prediction have non-zero coefficients; these coefficients are members of the *active set* [61]. Conversely, features in the *inactive set* are not relevant for prediction and thus have zero coefficients. In general, when we prefer models with small values in β we term this "regularization" [23]. Regularization and feature selection in general are important techniques in situations where we have many features and must use only those that are strong predictors of the target variable. Often penalties

in these contexts are written as norms, including the Lasso. A norm is calculated as:

$$\ell_\gamma = \left[\sum_{j=1}^{p} |\beta_j|^\gamma \right]^{\frac{1}{\gamma}}$$

where p is the number of features in the model [26]. With an ℓ_1 norm penalty, feature selection is performed as some coefficients are set to 0; this is the norm used for the Lasso. With an ℓ_2 norm penalty, coefficients are not set to zero and thus feature selection is not performed [62]; this norm is also known as the Euclidean norm.

Other prior information can be expressed similarly. In pain prediction, we will have multiple observations (i.e. instances) per patient. In our running example, we may have two patients:

$$X^{\{1\}} = \begin{bmatrix} -0.15 & 0.25 & -1.96 & -0.29 \\ -1.18 & -1.29 & -0.94 & 0.01 \\ 0.33 & -0.60 & 1.33 & 1.44 \\ 0.95 & 0.14 & -0.33 & -0.53 \\ 0.98 & 0.94 & 0.64 & -0.11 \end{bmatrix} \quad y^{\{1\}} = \begin{bmatrix} 0.27 \\ -8.16 \\ -0.77 \\ 3.64 \\ 5.24 \end{bmatrix}$$

$$X^{\{2\}} = \begin{bmatrix} -0.05 & -0.37 & -0.01 & -1.09 \\ -0.03 & -0.77 & -0.62 & -0.42 \\ 1.59 & 1.39 & -0.54 & 0.18 \\ 0.65 & 0.60 & -1.61 & -0.73 \\ 1.16 & 0.67 & -1.00 & -0.34 \end{bmatrix} \quad y^{\{2\}} = \begin{bmatrix} -2.07 \\ -4.75 \\ 3.49 \\ 0.07 \\ 3.55 \end{bmatrix}$$

where $X^{\{1\}}$ and $X^{\{2\}}$ are the design matrices for patients 1 and 2 respectively, and $y^{\{1\}}$ and $y^{\{2\}}$ are the target vectors for patients 1 and 2 respectively. In this example, the targets for patient 2 were generated using a different set of true coefficients β^* than patient 1:

$$\begin{bmatrix} 0 & 3 & 0 & 0 \end{bmatrix}^\mathsf{T}$$

For convenience, we may concatenate the respective vectors of coefficients for each patient as a single $p \times K$ matrix of coefficients Θ, where K is the number of patients: we call this the *coefficient matrix*. In our running example, the true coefficient matrix for patients 1 and 2 is written as:

$$\Theta^* = \begin{bmatrix} 3 & 0 \\ 3 & 3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

There are several potential assumptions we could make about the β generated by a predictive model for the two patients:

A1 A single β will work well for all patients

A2 Different patients will require different β

A3 Patients are mostly similar with respect to β, but some may be outliers

- Under A1 the Lasso is a good approach.
- Under A2 many separate Lassos are fine, provided that there is enough data per patient to train a predictive model.
- Under A3, neither approach fits the problem well, because A1 does not account for individual variation between patients and A2 does not leverage information shared between patients. This thesis explores and develops different methods for the context described in A3.

Previously, "dirty models" [31] designed for this purpose have decomposed the matrix of model coefficients Θ into a row-sparse matrix B and an elementwise sparse matrix S to model data of this structure. However, while this approach can represent global and local setting-specific effects, the results are not readily interpretable. Additionally, the dirty model this work is derived from does not scale as well as many other commonly used methods like the Lasso, which is an obstacle to big data analyses. In particular, for scalability we desire a model suitable for data from many patients (large K) but not large amounts of data about each individual patient (small n). The ANOVA (Analysis of Variance) technique in inferential statistics is another common way of detecting main effects (effects common to all patients) and interactions (individual effects) [49]. This is often used with categorical variables (variables with multiple levels that are not continuous), coded as $m-1$ binary dummy variables, where m is the number of levels of the categorical variable [22]. However, the ANOVA technique is concerned with detecting statistically significant differences, as opposed to building a model capable of accurate prediction. In this thesis, we treat variables in our design matrices as continuous numerical features. However, patient ID (or, more generally, setting ID) may be thought of as a categorical indicator variable.

This thesis addresses the following fundamental questions regarding the development of a dirty model suitable for predictive pain management:

- How can we adapt and design an MTL model suitable for healthcare contexts that gives us a more interpretable predictive model than existing models?
- How can we design such a model to scale to large data better than the existing dirty model?
- How do the recovered coefficients differ between our model and the dirty model, and does this affect interpretability?

These questions are addressed in the following major contributions of this thesis:

1. We propose a new "cleaner" dirty model, gLOP, for multitask learning.
2. We give a multitask learning algorithm that takes as input a set of design matrices of independent variables and a set of observation (dependent variable) vectors, and uses coordinate minimization and existing implementations of the standard Lasso to reach the global optimum of the objective function. We demonstrate that this coordinate minimization technique provides clear computational advantages over the standard dirty model.
3. We demonstrate that the interpretability of gLOP is superior to the dirty model, while still accurately representing the underlying structure of the data.

1.1 Motivation

This work represents an important step towards developing a system for automated predictive pain management (PPM). Significant challenges that exist for implementing and testing a full system for automated pain management include a lack of methods appropriate for this application and inadequate access to quality data for testing. Because the inputs to a future model for PPM will likely use features derived from waveforms (such as wavelets), trying to use all of the numerous possible features associated with these to model and predict analgesia requirements in real time would be prohibitively computationally expensive, and likely not feasible. Using feature selection techniques to choose the best features from all of those available would increase the speed of modelling and learning, and have the additional advantage of increased interpretability. As a first step, gLOP provides a method suitable for this purpose that in future can be used with patient data. This section provides a detailed motivation for developing methods for predictive

pain management in general, as well as additional challenges that exist for researchers in this area.

Properly managing pain in a recovery setting is a problem which has previously received little attention from the machine learning and general AI research communities. Mismanagement of pain following surgery can lead to very negative outcomes to patient health both in the short term and long term [11]. Pain is usually assessed and managed by clinicians; however, this is problematic because of the subjective nature of pain assessments. Potential knowledge deficits and improper execution of pain management techniques on behalf of the clinician may also cause inadequacies in the treatment of pain [18]. Better techniques to assist in assessing a patient's level of pain could improve clinician-administered treatments to manage pain. Patient-contolled analgesia (PCA) is a technique that allows patients to self-administer doses of analgesia when they are feeling pain. This technique is advantageous in that it does not rely on doctors to assess when a patient is in pain. However, PCA does rely on clinicians to impose restrictions on the dosage and timing between doses of analgesia. If these settings are miscalculated, patients may experience inadequate pain relief because of communication barriers with clinicians, or because of side effects from the medication [9]. Better prediction of PCA setting requirements would also have the potential to improve patient care.

Improperly managed pain may lead to longer recovery times and poorer patient outcomes [11]. For this reason, it is important to understand predictors of postoperative pain, as well as physiological and other potential indicators that may better reveal when a patient is experiencing levels of pain severe enough to be mitigated by pharmaceutical intervention. It has been shown in previous literature that doctors' estimates of pain are often unreliable [51]; for this reason, the ability to assess when a patient is experiencing pain, and to predict when a patient will experience pain in the future is particularly relevant for patients who are non-verbal, or who lack the ability to communicate with medical staff (such as neonatal patients) [18]. Assessing pain in non-verbal patients is usually attempted by using a behavioural pain scale [42], but there is no gold standard or general consensus on which pain inventory is the most appropriate given a patient's condition [3].

In addition to patient comfort, improper pain management in postsurgical settings can have serious consequences for patient health status outcomes and morbidity. Severe postoperative pain may be predictive of future chronic pain [32, 58]. Some physiological effects that may be exacerbated by poor postoperative pain management include immunosupression, tachycardia, hypertension, hyperglycaemia, decreased regional bloodflow and platelet aggregation [58]. Additionally, there has been recent research suggesting that experiencing severe pain activates neural fibres that can have long term effects on central nervous system function [32]. Because of this, not only is it crucial to treat pain effectively, it may be

better to proactively administer analgesia or anaesthetic to prevent such pathways from being activated even before surgery takes place [58].

While pain is usually undertreated, it is also important to consider the consequences of overusing specific types of analgesia on a patient. In particular, opioid use may result in either tolerance (desensitization of antinocioceptive pathways) or sensitization to painful stimuli (Opioid Induced Hyperalgesia (OIH); an increase in pain caused by upregulation of pronocioceptive pathways). Both of these result in an increased amount of opioid needed to treat the same amount of pain [58]. To mitigate these negative effects, it is possible to combine or replace opiod usage with other drugs that can either reduce the amount of opioid to be used or relieve the effects of OIH; dexmedetomidine and ketamine are respective examples of such drugs that may be used for treatment.

In general, physicians and nurses (who are often the primary administrator of pain management) are limited in their ability to manage pain consistently and effectively due to the subjective nature of the assessment tools available to them [18]. Additional difficulties with pain management may arise due to discrepancies between the knowledge of practitioners and the actual quality of care. In a study of nurses' knowledge in relation to pain management practices with surgical patients assigned to them it was found that all nurses exhibited knowledge deficits regarding pain management regardless of how positively they rated their own knowledge of pain management practices [54]. In this study, nurses' knowledge scores were unrelated to patients' perceptions of quality of pain management. Additionally, patients experiencing moderate to severe levels of pain were not treated using the full amount of analgesia prescribed to them; these patients were given only 47% of the analgesia allotted to them. A later study was conducted similarly, but was specifically focused on nurses' perceived abilities to make judgements of pain with various assessment tools and the effect of these tools on their clinical judgement [56]. This study also attempted to implement a standardized procedure for the assessment of pain and sedation in patients, followed by focus groups to discuss the nurses' opinions on the protocol. While nurses thought the tools were helpful in some sense, they still expressed reliance on their own particular methods of judging patient condition; this standardization procedure was perceived by the nurses to limit their thought processes. Despite this, the guidelines facilitated deeper analysis of patient condition as it required a quantitative, rather than qualitative, measure of patient condition [56]. Having practitioners adopt standardized regimes is a crucial step in reducing biases that contribute to lower standards of patient care.

A popular method of dealing with challenges associated with pain assessment and dosing had been the administration of patient-controlled analgesia (PCA). This is advantageous in some respects as it allows the patient to control the intake of pharmaceuticals according

to their specific pharmacokinetic [46] and pharmacogenetic [47] characteristics [20]. The frequency and number of unsuccessful attempts at administering PCA (because of having reached the dosage threshold limit) is also thought to be a better indicator of the amount of pain a patient is in than verbal pain assessments [9]. PCA has been used increasingly for eligible patients in the recent past; in one clinic, the incidence of PCA usage was found to have increased by more than three times over a 10 year period [9]. In this same study, it was found that morphine consumption among individual patients had decreased over this time span. Despite the increased usage of PCA, the refinement of administration techniques and patient education, and increased ratings of patient satisfaction, the audit performed by Cheung et al. [9] found no clinically significant improvement in pain treatment over this period of time. These ratings of satisfaction may not be reflective of general improvements in the quality of postoperative pain management for a number of reasons. Patients may be reluctant to complain about the quality of their treatment to caregivers, and expectations of postoperative pain management quality may influence patients' feelings of satisfaction [9]. Despite patients having the ability to titrate their own dosage scheduling, the use of morphine PCA may cause postoperative nausea and vomiting (PONV), which may lead patients to under-treat pain in order to minimize side effects. Problematically, if pain is left untreated because the patient is reluctant to administer PCA, side effects of nausea and vomiting may also occur [9]. Although efforts to refine PCA in the context of acute pain services have been made, it is apparent that additional improvements to this method of treatment are necessary if postoperative pain is to be managed effectively.

Because of the prevalence of opioid use for treating acute pain in postsurgical settings, it is important to understand both the consequences of its effects on patient recovery in relation to other patient factors. An important priority in exploring this issue is to devise and assess more effective methods of determining when patients are in pain and predicting when a patient will require administration of analgesia in the future. The use of more statistically rigorous feature selection methods in this context is a necessary first step to building models which are capable of predicting when a patient may require analgesia in the future.

In this thesis, we present a novel statistical method using predictive models as a contribution towards a full solution to the problem of PPM. Predictive models have been extensively studied in machine learning, but this is a novel context for such. Our model can be used in many applications appropriate for multitask learning and dirty models in particular (such as classification of handwritten digits written by many different individuals [31]). A significant contribution of this thesis is a novel machine learning tool applicable to many different problem domains.

1.2 Thesis statement

This thesis addresses the development of a predictive model suitable for use in predictive pain management. As discussed above, such a model must be interpretable, scalable and capable of representing dirty data, i.e. data where both individual model contributions and global population-wide effects are present. This thesis addresses each one of these criteria in the following related research questions:

1. How can we decompose the matrix of coefficients Θ to reflect individual differences, but still capture global effects in a way that is easily interpretable?
2. What penalties are most suitable for each of the decomposed coefficient matrices to maximize predictive power as well as the accuracy of the recovered underlying structure of the data?
3. Does coordinate minimization using existing implementations of the Lasso provide superior computational performance over the standard dirty model?
4. How do the recovered values in the coefficient matrices change when β is decomposed differently than in the standard dirty model?

1.3 Organization

Chapter 2 of this thesis summarizes previous work in feature selection, single task learning and multitask learning with clean and dirty models.

Chapter 3 of this thesis contains the problem formulation of gLOP and descriptions of the synthetic data generated, our cross-validation procedure and experimental results comparing gLOP with the standard dirty model.

Chapter 4 provides several future directions related to this work that we will pursue to improve the model for use on real-world data.

Chapter 5 summarizes the contributions and findings of the thesis.

Appendix A provides a more detailed background on the motivation for our application of predictive pain management, and summarizes previous work that has been done in this area. This section also provides a basis for future work in this area.

Appendix B provides an overview of preliminary analyses and challenges experienced with the MIMIC-II data set.

Chapter 2

Background

Feature selection, or the identification of factors relevant for predicting a designated response in classification or regression problems, is an important consideration especially when dealing with high-dimensional data. This is particularly true when potential predictors are more numerous than observations available for training a predictive model, as is often the case with clinical data. Often feature selection is done using data drawn from a single distribution, trained to model one particular task; this is known as single-task learning (STL) [7]. A classic example of a single-task learning model is regression, where we have a binary $n \times 1$ response variable y, an $n \times p$ design matrix of predictor variables X, and a $p \times 1$ vector of coefficients. In the regression problem, we minimize the residual sum of squares (RSS) to obtain ordinary least squares (OLS) estimates [23]. For classification, the RSS loss function is typically replaced by logistic loss (giving logistic regression, used when the target variable is binary or categorical) [23] or hinge loss (giving the Support Vector Machine [23]). However, several drawbacks of this method prompted the production of more robust models such as the Lasso [50]. One drawback of using OLS estimates for prediction is low prediction accuracy when $p >> n$, due to the large variance of estimates, despite low bias. This problem may be ameliorated by shrinking the size of coefficients in the model, which is an attractive property of ridge regression. Setting coefficients to zero may also improve prediction accuracy; both of these strategies may reduce variance of predicted values and thus increase overall accuracy of predictions, but also sacrifice bias. Another drawback of using OLS estimates is that all features are included in the model, no matter how strong or weak their prediction effects might be. This is problematic for interpretation, when we would like to identify a subset of the few strongest predictors [23]. Subset selection offers a solution to the issue of interpretability as it eliminates weak predictors from the model; this may be accomplished using a variety of methods such as

best-subset selection and the Lasso, which has the added benefit of shrinking non-zero estimates to reduce variance. Similarly, the group Lasso allows groupings of predictors to be selected sparsely, which is an advantage when dealing with dummy variables or when one has apriori knowledge of predictors that should be included in the model as a group[59]. In this chapter, these models will be described in detail, as they are the forerunners of models that will be used to address the problem of feature selection for predictive pain management. Additionally, the Composite Absolute Penalties (CAP) family of penalties will be described as a generalization of the Lasso and group Lasso.

2.1 Feature Selection

2.1.1 Subset Selection

Many methods exist for selecting subsets of the strongest predictive variables; more complicated methods involve k-fold cross validation to select the best subset [23], but they will not be discussed in this section. Best subset selection is a method of subset selection that examines all possible subsets of size k for $k \in 0, 1, ..., p$, where p is the number of predictors. The subset chosen is the one which yields the smallest residual sum of squares; other criteria may be used to determine which subset is the most desirable. Often when sparsity or model parsimony is desired, the one chosen will be the smallest subset with the lowest model error, which may require a tradeoff between variance and bias levels. While this method is obviously not feasible for situations in which p is large, when $p \leq 40$ this method is possible when used with the *leaps and bounds* procedure [23].

Another subset selection technique that may be used for data sets with larger numbers of features is Forward-stepwise regression, which employs a greedy approach to find the most desirable subset. The end result is a k-indexed sequence of models with an increasing number of factors. This technique starts with a model only including the intercept, and at each stage adds the best-fitting predictor to the model; this produces the nested sequence of models described above. Advantages of Forward-stepwise regression over best-subset selection include the lower computational overhead required to perform this technique. Because not all possible subsets must be explored, Forward-stepwise regression can be done even when $p >> N$, as is often the case when dealing with real-world data, particularly in the medical domain. Another advantage of this technique is that is more constrained than best-subset, and thus has lower variance but at the cost of more bias. Backward-stepwise regression works similarly (and has similar advantages), but instead starts with all features included in the model and at each step, removes the feature that is the weakest predictor

(the feature with the lowest z-score). Unfortunately this technique cannot be used when $p > N$, where N is the total number of observations in the data set, and thus is not suitable for high-dimensional problems (i.e. problems where the number of features is much larger than the number of observations or data points). Some implementations of these step methods can also handle groups of predictors, such as dummy variables representing levels of a categorical feature [23].

A similar technique to forward stepwise regression is forward stagewise regression, which is more constrained than either of the stepwise approaches. In forward stagewise regression, predictors are all centered with coefficients of 0, and the only term in the model is the intercept, which is computed as the mean of the observations $\hat{y}$. The algorithm chooses the predictor most correlated with the residual at each step, and once the coefficient of the linear regression of the variable is computed, it adds this to the variable's model coefficient. The procedure stops when no more variables are correlated to the current residual. While this is very inefficient, since the procedure may take $>> p$ steps to find the optimal fit due to the lack of adjustment of other predictor coefficients at each step like in stepwise procedures. However, this slow fitting procedure may be advantageous when dealing with high-dimensional problems [23].

Subset selection techniques can also be incorporated directly into classifiers or other learning algorithms using wrapper and filter techniques [33]. Filter techniques work independently of the induction algorithm; subset selection is performed as a preprocessing step on the input features, and the chosen subset becomes the new set of input features for the algorithm. In contrast, wrapper techniques work by using the induction algorithm itself to evaluate each subset. In this technique, the algorithm functions as a black box; it conducts a search for the subset with the best classifier accuracy in the state space using a heuristic/evaluation function and a stopping criterion. Usually the heuristic/evaluation function used is k-fold cross validation repeated on all features, the initial state is the empty set of features, and the state is a p-length boolean vector representing whether or not the feature is included in the model. Two common search techniques may be used: hill-climbing (greedy), or best-first search. Because hill-climbing gets stuck in local optima very often, usually the technique of choice is best-first search [33].

While subset selection offers a solution to the problem of interpretability when using OLS estimates, several disadvantages of subset selection prompted the development of shrinkage/selection techniques such as the Lasso. The biggest drawback is the lack of model stability when using subset selection. Even small variations in the data may produce very different subsets of selected variables. This instability may in turn cause prediction accuracy to suffer [50].

While subset selection works to select the best features to use for modelling, other techniques exist that perform modelling and feature selection simultaneously via regularization. This can be done by penalizing a norm of the coefficients, as in ridge regression, bridge regression and the Lasso. Depending on the norm chosen, this may be used to perform feature selection. Other methods use a hybrid penalty that combines two norms to perform feature selection at the factor level (i.e. selecting groups of variables, such as dummy variables corresponding to the same feature, instead of individual variables), which is useful in cases where groups of features are known apriori. While we likely would not be able to gather enough data from any one patient to perform feature selection using single task methods for predictive healthcare applications, single task learning methods provide a basis for MTL methods that we will use in the development of gLOP, which has attractive characteristics for healthcare applications. In the remainder of this chapter, we will describe single task modelling and feature selection methods to provide background for subsequently described methods for MTL.

2.1.2 Ridge, Lasso, and Bridge Regression

Ridge regression [25] was introduced to reduce variance in situations where predictors in the context of multiple linear regression are highly correlated, as is often the case when working with real-world data. Correlations between features may introduce bias to the least squares estimates, which may then suffer from instability and inflation. Ridge regression is a modification of Least Squares; this procedure shrinks coefficients by penalizing the ℓ_2 (Euclidean) norm of the coefficients β produced via linear regression [62]. The shrinkage is encouraged by imposing higher penalties on higher coefficients (so that these values are reduced), and smaller penalties for smaller coefficients [61].

The Least Absolute Shrinkage and Selection Operator (Lasso) and its many variations, including the group Lasso (to be subsequently described), were developed to provide methods of feature selection that were capable of shrinking coefficients to decrease variance, and setting other coefficients to 0 to encourage sparsity and select only the most relevant features for inclusion in a given model. These techniques are applicable to many different methods including OLS and linear regression. Given an $n \times p$ design matrix of predictor variables X, and a binary $n \times 1$ response variable y, the Lasso is defined by Tibshirani [50] as:

$$\hat{\beta}_\lambda = \arg\min_\beta \{\tfrac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1\}$$

where λ controls the amount of shrinkage of the estimates. When $\lambda = 0$, the model is unpenalized with all features present; higher values of λ will cause more estimates to

be shrunk towards 0, giving a sparser model. It is assumed that all entries of X are standardized. Unlike ridge regression which uses the ℓ_2 norm as a penalty and scales coefficients, the Lasso uses the ℓ_1 norm, translates coefficients by a constant factor and truncates them at 0, allowing factors to be removed from the model [50].

Finally, Bridge regression [16] is a generalization of both ridge regression and the Lasso; it allows β to be penalized by penalties other than the Euclidean norm or the ℓ_1 norm. Similar to both of these methods, the bridge estimate may be written as

$$\hat{\beta}_\lambda = \arg\min_\beta \{\tfrac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_\gamma\}$$

where the new penalty used is the ℓ_γ norm [62]:

$$\|\beta\|_\gamma = (\sum_{j=1}^{p} |\beta_j|^\gamma)^{1/\gamma}.$$

This variable penalty allows bridge estimates to be suitable for regularization with or without feature selection; when $0 < \gamma \leq 1$, bridge performs feature selection (as evidenced by the behaviour of the Lasso, where $\gamma = 1$). In contrast, bridge shrinks coefficients without encouraging sparsity when $\gamma > 1$ (as in ridge regression where $\gamma = 2$).

2.1.3 Group Lasso and CAP Family Penalties

Unlike the Lasso and bridge regression in general, the group Lasso allows for pre-specified groups of variables to be considered together for inclusion or exclusion in a given model. For example, categorical variables are often encoded in the form of $d - 1$ binary dummy variables, where d is the number of levels in the categorical variable. It follows that all components of a single factor, or all dummy variables corresponding to one categorical variable, form a natural grouping and should enter the model together as a group.

For this reason (again, unlike the Lasso) the group Lasso was designed to be invariant to linear feature transformations within groups, as Yuan [59] determined that factor selection should not depend on factor representation. Given an $n \times p$ design matrix of predictor variables X, a binary $n \times 1$ response variable y, and a $k \times 1$ vector G of group indicator variables, the group Lasso is defined by Yuan [59] as:

$$\hat{\beta}_\lambda = \arg\min_\beta \{\tfrac{1}{2}\|y - X\beta\|_2^2 + \lambda \sum_{k=1}^{K} \|\beta_{G_k}\|_2\}$$

Instead of applying an ℓ_1 penalty, the penalty for the group Lasso is a hybrid between the Lasso's ℓ_1 penalty, which encourages individual factors to be selected sparsely, and the ℓ_2 penalty of ridge regression, which does not encourage sparsity. The resulting penalty facilitates sparse factor level feature selection. This property is apparent when looking at the geometry of each penalty, considering the case when two factors are present: the ℓ_1 penalty treats each coordinate direction differently than the others, and thus is shaped like two pyramids stacked base to base. Conversely, the ℓ_2 penalty is spherical in form as all coordinate directions are treated the same. The group Lasso penalty treats coordinate directions for members of the same group similarly, but not coordinate directions between factors; as a result, this penalty is shaped like two cones stacked base to base.

In general, bridge regression and the group Lasso are members of the Composite Absolute Penalties (CAP) family of penalties, which may be used in various cases of hierarchical and group-based feature selection. General CAP family penalties allow for combinations of different norms besides ℓ_1 and ℓ_2 (as seen in the previously described models) to be used; attractively, optimization is easy when using CAP penalties, as these penalties are convex, provided that the norms used for the penalties are also convex. CAP penalties also extend the framework of the group Lasso such that overlapping groups of features may be used for hierarchical feature selection [62].

Intuitively, CAP penalties work by imposing different norms on the coefficients of different groups of variables, and imposing an overall norm that performs selection across groups. For hierarchies, this causes coefficients to enter the model in a specific order. As defined by Zhao et al. [61], construction of a general CAP penalty works as follows: for a design matrix X, response vector y and vector of model coefficients β, we designate groups G_k where $k = 1, ..., K$, maintaining the natural grouping structure of the variables. These groups may be non-overlapping, or overlapping which may reflect a hierarchy of variables. For each of the K groups, we create vectors comprised of the regressors and their coefficients denoted $\beta_{G_k} = (\beta_j)_{j \in G_k}$. We take the norm $N_k = \|\beta_{G_k}\|_{\gamma_k}$ of these vectors, and form an aggregate K-dimensional vector $N = (N_1, ..., N_k)$ using the computed norms. The CAP penalty is then computed as:

$$N_k = \|N\|_{\gamma_0}^{\gamma_0} = \sum_k |N_k|^{\gamma_0},$$

where γ_0 is a predefined norm taken across all groups. Finally, we define the corresponding CAP estimate as a function of λ (the regularization parameter):

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \sum_i L(Y_i, X_i, \beta) + \lambda \cdot T(\beta),$$

where $T(\beta)$ is the penalty function and $L(Y_i, X_i, \beta)$ is the loss function representing the model's goodness of fit, which is usually dependent on the model application. In order to facilitate sparsity across groups we select $\gamma_0 = 1$, and to ensure that features within a group enter the model at the same time we select $\gamma > 1$.

While many different loss functions may be used in the CAP equation, defining the loss L as the log-likelihood of the data given the vector of coefficients β allows for a Bayesian interpretation of CAP penalization, as this type of penalized estimation is connected to Maximum a Posteriori (MAP) estimates. In this case, the penalized estimates correspond to the MAP coefficient estimates, and the penalty T may be seen as the log of an *a priori* probability function; this makes intuitive sense, as the penalty function in the estimation procedure prefers solutions that are more likely under the prior, which is dependent on γ [61]. In the Bayesian view of ridge regression, the data is assumed to have a gaussian distribution given β, with a gaussian prior on β. Conversely, in bridge regression the distribution of y remains gaussian, but the prior changes depending on the selection of γ. When considering the density of these priors, priors with $\gamma < 2$ have a cusp at 0, which may be seen to facilitate sparsity; when $\gamma \geq 2$, there is no cusp apparent in the density of the prior, and consequently sparsity is not encouraged. The density of the assumed a priori distribution in this context is defined as:

$$f(\beta) = C^1_{\gamma_0,\gamma} \exp\{-\sum_{k=1}^{K} (\|\beta_{G_k}\|_{\gamma_k})^{\gamma_0}\},$$

where the constant $C^1_{\gamma_0,\gamma}$ allows the density to integrate to 1 [61].

Using this joint distribution, a high level summary of the structure promoted by the composite absolute penalty is as follows: the between group vector of coefficients N is i.i.d. sampled from the density function f, where $f_{\gamma_0}(x) \propto \exp(x^{\gamma_0})$, in which γ_0 acts similarly across groups to the γ used in bridge regression does across individual factors. When $\gamma_0 \leq 1$, sparsity is facilitated in that some group norms are set to 0; conversely, when $1 < \gamma_0 < 2$, group norms are encouraged to be dissimilar. Finally, when $2 < \gamma_0 < \infty$, group norms are encouraged to be similar. Following the sampling of N from f_{γ_0}, the scaled coefficients are defined as $\frac{\bar{\beta}_{G_k}}{\|\bar{\beta}_{G_k}\|_{\gamma_k}}$, where k is the group index. These scaled coefficients are independently and uniformly distributed on the L_{γ_k} norm-defined unit sphere, if it is assumed that the groups do not overlap. Because of this, smaller values of γ_k promote coefficients within group k to concentrate on the coordinate axis, while larger values promote coefficients within group k to concentrate on the diagonals [61].

2.2 Multitask Learning

Often feature selection techniques like the Lasso and group Lasso work under the assumption that adequate amounts of data will be provided for the purposes of training, and that each training observation comes from the same underlying distribution as all other training observations. However, in many medical domains including PPM it is infeasible to collect enough data for any one patient to adequately train a predictive model. Consider the case where a predictive model is trained for each individual patient, without the addition of aggregate data from any other other patient. Given enough data, it would be reasonable to assume that certain useful predictors would be common between all patient models for predicting a given outcome; given that each patient has an individual model, it is also possible that these common predictive features would have different coefficients for each individual. In addition to the features common among all patients, it is likely that individual models might include features that are useful for predicting an outcome for that particular individual, but that some such features are not useful for predicting the same outcome for other individuals.

In order to train a general model suitable for all patients, aggregate data must be leveraged if individual patients themselves do not have sufficient data to train a model. However, the Lasso and group Lasso are not designed to take advantage of the additional information about which underlying distribution a given observation is derived from. Multitask learning (MTL) provides a framework for taking advantage of this information via inductive transfer in order to facilitate learning and generalization and prevent overfitting to any one task. Because our real-world PPM data will likely have these challenging characteristics (many patients, and very little data per patient), we hypothesize that MTL will provide increased performance over STL. MTL methods that have been used in previous research with similar types of challenging data are described below.

2.2.1 Backpropagation MTL

Caruana [7] provided the original framework for MTL in the context of learning using backpropagation neural networks. The intuition behind MTL using backpropagation neural networks is that learning with multiple tasks can improve generalizability in a number of ways, although Caruana primarily focuses on improvements in this area stemming from the relatedness of tasks in the same backpropagation net. Other potential causes for improvement in model generalizability discussed in Caruana's paper include the aggregate gradient used between tasks for optimization; if tasks are not very correlated, this aggregate gradient appears as noise to other tasks in the same learner. Because of this noise

in the classifier, generalization performance is improved. Another possible reason for improvement in this area for backpropagation neural nets is the change in weight updating dynamics caused by adding multiple tasks to a network. Having additional tasks causes the learning rate between the inputs and the hidden layer to increase, versus the learning rate between the hidden and output layers. It is possible that this increased learning rate in the input-to-hidden layer is responsible for improved generalizability in this context. The final two possible reasons cited for this increase in performance are the reduced network capacity derived from the hidden layer shared between all tasks, and the reduced likelihood of getting stuck in local minima that is common in single task backpropagation learning. In the case of multitask learning where a single task might be associated with a local minimum, it is possible that other tasks learned using the same network have the ability to push it out of that minimum [7].

The relatedness of tasks is an important consideration in MTL, as improvements in performance may rely heavily on how tasks are related to one another within the learner. One potential definition of relatedness in the context of MTL is described by Caruana as follows: a task is related to another task if and only if there is better generalizability on the main task when the additional task is learned in parallel. The disadvantage of this definition is that it doesn't specify what type of learning is taking place in this context, which may be an important consideration. Instead of a single definition of what makes a particular task related to another, the author describes a number of characteristics of tasks that are related to one another. The first of these is that related tasks are not correlated, in the sense of their individual signals being correlated (an example is shown below). More important is the correlation between task representations; the level of representation between tasks must be correlated, but the outputs in the network between two tasks do not have to be correlated. An example given in the context of backpropagation MTL is a situation with two synthetic tasks F_1 and F_2 which are functions of inputs A and B. Consider the case where $F_1(A, B) = \text{sigmoid}(A + B)$ and $F_2(A, B) = \text{sigmoid}(A - B)$. In this example, because $A + B$ and $A - B$ are not correlated, F_1 and F_2 are not correlated. However, the representation of these tasks is correlated, making the two tasks related to one another [7].

Another feature underlying task relatedness is that input features must be shared between related tasks, which is the type of relatedness we expect to see in our application area. If the feature sets of two tasks are completely disjoint, no MTL can take place as no weights would be shared in the hidden layer of a backprop net. In this sense, relatedness between two tasks may be measured by the amount of overlap in inputs that the tasks share; however, having shared inputs does not guarantee that two tasks will be related. Instead, tasks must also have shared or similar functions of input features to allow suc-

cessful MTL to take place. In addition to these characteristics, heuristics may also be used to determine task relatedness. These heuristics focus on defining what, if any, intertask relationships can be exploited by a given learning algorithm, and in particular, by a good MTL algorithm. Finally, Caruana notes that even when tasks are related by these criteria, MTL algorithms might fail to benefit from the additional information derived by training the learner on the related auxiliary task. When intertask relationships cannot be exploited in this way, then future work remains to be done in improving the algorithm to have the capability to learn from the related task.

In order to demonstrate the efficacy of this method, Caruana explored the use of backpropagation MTL in the context of predicting pneumonia severity using the Medis Dataset [15]. Inputs to the network included basic demographic and clinical information collected prior to hospitalization, while the main output of the network was patient mortality (stratifying patients by pneumonia risk; the main task), and extra outputs of the network consisted of 35 future laboratory tests undertaken after the patient was admitted to the hospital (the auxiliary tasks). It was noted that it would be ideal to use these laboratory tests as part of the input layer, but it would be unreasonable to use them for prediction in this case, as they are not collected prior to hospitalization. The MTL was contrasted with single task learning (STL) using backpropagation using SSE on repeatedly re-estimated tasks (Rankprop). The network was trained on randomly drawn sets of 1000 patients patients from the database. It was found that MTL statistically significantly reduced prediction error compared to STL; although the absolute value of this improvement appeared to be small, it is large enough to be of practical importance in the medical domain.

2.2.2 Clean and Dirty Models for MTL

While MTL has been shown to work in backpropagation neural networks, many others have explored its use in more conventional implementations of penalized regression and regularization such as the CAP family penalties discussed previously [31, 34]. A particular favourite CAP penalty used in many such studies is the (ℓ_1, ℓ_γ) norm which imposes a Lasso-type penalty across group norms to facilitate sparsity at the factor level, and a within group norm that does not facilitate sparsity, such as the ridge penalty. The ℓ_∞ norm also has attractive properties as a within-group penalty that does not facilitate sparsity; given some structural constraints, minimum performance bounds and guarantees of signed support have been established for this method [38]. In order to maintain separation between tasks for learning in this setting, an aggregate matrix of all of the tasks is constructed in a block fashion. Given K tasks, design matrices $X^{\{1\}}, ..., X^{\{K\}}$, and observation vectors $y^{\{1\}}, ..., y^{\{K\}}$, the block matrix is constructed as follows:

$$X = \begin{bmatrix} X^{\{1\}} & 0 & \cdots & 0 \\ 0 & X^{\{2\}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & X^{\{K\}} \end{bmatrix}.$$

This corresponds to the clean model structure wherein features are exclusively block- or row-sparse. Another example of a clean model is where data structure is assumed to be exclusively elementwise-sparse.

Related work has also focused on creating MTL models for "dirty data": data with features that may not be shared completely across all tasks [31]. Data of this type must be approached carefully because if there is insufficient overlap of features between the tasks, or if the parameter values for the shared tasks vary widely, then the intuitive approach (block regularization) may do more poorly than the Lasso which does not take advantage of additional task information [31]. Instead of assuming all tasks are equally related, Jalali et al.(2010) [31] outlined a "dirty model" to leverage existing parameter overlap in dirty data, but penalize when the overlap is insufficient for the model to take advantage of task relatedness. Parameters may differ between tasks in this dirty model. In relation to our running example described in the introductory chapter, this corresponds to the situation in which most patients may share a predictive model, but some patients require different models. The optimization problem for the dirty model is as follows [1]:

$$\arg\min_{B,S} \sum_{k=1}^{K} \|y^{\{k\}} - X^{\{k\}}(B^{\{k\}} + S^{\{k\}})\|_2^2 + \lambda_B \|B\|_{1,\infty} + \lambda_S \|S\|_{1,1} \tag{2.1}$$

Inputs to the dirty model are $K > 1$ response variables and a common set of p features, with n samples per task. Usually "clean" models either use a row-sparse structure [38] or an elementwise-sparse structure [50], but Jalali's model combines the two structural forms. The dirty model uses the $\ell_{1,\infty}$ norm to estimate the sum of parameter matrices B, a block-structured row-sparse matrix, where the $\ell_{1,\infty}$ norm is computed for any matrix M as [31]:

$$\|M\|_{1,\infty} := \sum_j \|M_j\|_\infty \tag{2.2}$$

where:

$$\|M_j\|_\infty := \max_k |m_{k,j}|. \tag{2.3}$$

[1] Here we give the objective function as defined in the code provided by Jalali. It differs from the objective stated in the paper by a factor of $\frac{1}{2n}$ applied to the squared loss term.

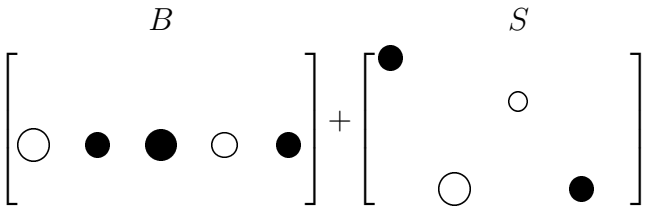

Figure 2.1: A sample dirty model structure. Filled circles represent positive coefficients, empty circles represent negative coefficients.

The secondary parameter matrix S is an element-wise sparse matrix that is penalized using the $\ell_{1,1}$ norm, or the sum of the absolute values of the elements in the matrix [31]:

$$\|M\|_{1,1} := \sum_{j,k} |m_{k,j}|. \tag{2.4}$$

Sample structures for B and S in the dirty model are shown in Figure 2.1, where both B and S are $p \times K$ matrices.

This work relies heavily on previous work proving bounds and other performance results for the $\ell_{1,\infty}$ norm, thus the task and design matrix setup follows the formatting outlined by Negahban (2008) [38]. Following this format, the authors assumed certain conditions common for proving performance consistency of the Lasso. The first of these is a constraint on the minimum eigenvalue of the design matrix to prevent too much dependence between members of the active set (the set of non-zero coefficients) [39]. Similarly, the second of these is a Maximum Boundedness condition on the maximum eigenvalue of the design matrix. Essentially, the these two conditions are meant to ensure that the matrix containing only the active set of variables is full rank. Additionally, these conditions constrain the size of the active set to be bounded by the minimum of n and p. The third condition that the authors imposed is an Incoherence Condition, which constrains correlations between variables of the active set and the inactive set: to satisfy this condition, there cannot be high correlations between columns of the design matrix that are in the model (members of the active set) and columns of the design matrix that are not in the model (members not in the active set). Formal conditions exist that imply these criteria; for additional details see [31].

Assuming these conditions hold, Jalali and his coauthors showed that, given enough data, with high probability the solution to this convex optimization problem has a unique optimum, the estimated set $\hat{\Theta} = (B + S)$ of supports between the tasks has no falsely

included parameters, and the ℓ_∞ error is bounded. Additionally, it was shown that the sign of the true support of the parameter $\bar{\Theta}$ will be the same as the sign of the true support of the estimate $\hat{\Theta}$, provided that the absolute value of each element of $\bar{\Theta}$ is greater than a threshold ensuring that the coefficient is large enough to distinguish from noise (i.e. the true effect of each feature in the predictive model will be correctly identified).

These results were demonstrated experimentally, in a two-task setup with synthetic i.i.d. Gaussian design matrices with zero mean and unit variance. The non-zero entries of $\bar{\Theta}$ were randomly chosen to each of 100 trial runs with a fixed set of matrices and similarly were given values with zero mean and unit variance. These values were used to generate n samples per task, and the penalty regularizer coefficients were chosen via cross validation. To test the performance of the dirty model, the authors used five different "overlap" ratios between the tasks (i.e. for a low overlap ratio, few features would be shared between tasks, whereas for a high overlap ratio many features would be shared between the tasks), and three different numbers of features for the tasks. Recovering the correct signed support was designated as the criterion for success. As predicted by the authors, in this experimental setup the dirty model performs better than both of the clean methods which use either solely row-sparse or elementwise-sparse structures.

2.3 Implications for Predictive Pain Management

The MTL method described by Jalali et al. [31] provides a practical starting point for the proposed application of predictive pain management as the structure will allow us to explore the effects of different data structures and methods of condensing data between patients and events. Because predictive pain management requires working with real data, the focus in the described studies on working with dirty data is useful for the current work, since we must represent both individual effects and population-wide global effects. In the dirty model, B represents effects across settings (shared features), and S represents setting-specific effects (non-shared individual features).

Chapter 3

gLOP

Decomposing the Θ matrix into matrices B and S as Jalali does [31] allows us to more accurately represent the underlying structure of dirty data. However, this paradigm may not be optimal for interpretability, as coefficients for a single feature in B are not required to be the same for all settings. Without requiring the model to have the same coefficient values for each feature across all settings, one cannot interpret individual effects captured in the local model the same way in relation to each setting, as the effects would also vary across settings in the global model. Applying the $\ell_{1,\infty}$ penalty to B encourages coefficients to converge to a similar absolute value, but it does not *require* the values to be the same, even though this is implied by the diagrams in work by Motamedvarziri [37]. Additionally, Motamedvarziri's diagrams [37] imply that for each feature, the paradigm uses the same coefficient value in B that is best for all individual settings, and that coefficients in S may overlap with these to reduce or increase the impact of the feature for a particular setting. This conflicts with diagrams given by Jalali [31], who authored the paper on dirty models that we are using as the basis for the current work. Our experimental results also suggest that overlap of the nature suggested by Motamedvarziri [37] seldom occurs in practice.

While having a different coefficient for each setting may be a more accurate representation of the underlying data structure, it prohibits one from being able to interpret the coefficients of B as "global" to all settings. This is the case in the Jalali paradigm, as the models for different individuals would have different contributions from B as well as S. A common technique in inferential statistics that allows one to determine common and individual characteristics of patients in healthcare applications is the ANalysis Of VAriance (ANOVA) test in combination with post-hoc tests, possibly with a correction for multiple comparisons. In the ANOVA setting, common patient characteristics correspond to main effects in our problem. Individual patient characteristics would comprise the interactions

in this scenario, obtained via post-hoc analyses such as the Tukey HSD (Honestly Significant Different) test or Fisher's LSD (Least Significant Different) test which determine individual interactions between features in the model. A significant advantage of using this type of inferential test is that it is relatively easy to interpret, particularly for clinicians and other members of health care teams who may not be familiar with penalized regression and other machine learning techniques. Another substantial advantage of using more simple techniques in inferential statistics is the lowered computational demands of these paradigms. Using analysis techniques that are less time consuming is preferable, particularly when attempting to process data in real time or according to deadlines related to patient treatment schedules.

However, the goal of any ANOVA (or other hypothesis testing framework) is not to build a model is capable of accurate prediction, but to identify coefficients that are non-zero at a given level of statistical significance. Since our goal is to construct good predictive models, we will not use the ANOVA, but will combine the underlying principles behind the interpretation of ANOVA results and the structure of dirty models. By doing this, we get the best of both worlds: an interpretable model capable of prediction that can accurately recover the underlying structure of the data. In the following section, we first propose a new "cleaner" dirty model, gLOP, for multitask learning. Following this, we give an algorithm that uses coordinate minimization and existing implementations of the standard Lasso to reach the global optimum. We subsequently demonstrate that this coordinate minimization technique provides clear computational advantages over the standard dirty model. Finally, we present experimental results demonstrating that the interpretability of gLOP is superior to the dirty model, while still accurately representing the underlying structure of the data.

3.1 Problem Formulation

Like Jalali, we decompose the Θ matrix into two parameters, but our decomposition uses a *vector* g and a matrix L. The $p \times 1$ vector g contains global coefficients that apply to all settings, analogous to main effects in the inferential statistics paradigm. The $p \times K$ matrix L contains local coefficients that apply only to their specific settings, which are analogous to the interaction terms in the inferential statistics paradigm. This makes our model more interpretable, as main effects are clearly distinguishable from individual effects, i.e. effects that are setting-specific. We use the ℓ_1 norm instead of the $\ell_{1,\infty}$ norm used by Jalali, since g is a vector instead of a matrix (analogous to B) in our paradigm, and it would be undesirable for our purposes to force the coefficients in g to converge to a common absolute

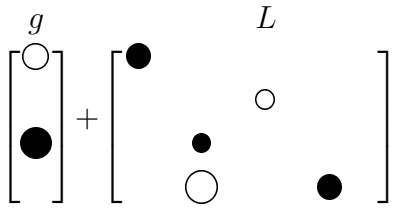

Figure 3.1: A sample gLOP structure. Filled circles represent positive coefficients, empty circles represent negative coefficients.

value. Our objective is:

$$\min_{g,L} \sum_{k=1}^{K} \tfrac{1}{2}\|y^{\{k\}} - X^{\{k\}}(g + L_k)\|_2^2 + \lambda_g \|g\|_1 + \lambda_L \|L\|_{1,1}. \tag{3.1}$$

This solves the interpretability problem of Jalali's model, as coefficients in g are fixed across groups. A comparative diagram to Figure 2.1 illustrates a sample gLOP structure where g is a $p \times 1$ vector and L is a $p \times K$ matrix (Figure 3.1). Note that if we set $B_k = a \forall k$ and set $S = L$, the objectives are identical.

3.2 Optimization Approach

We now present an algorithm for optimizing our objective shown in Expression 3.1 based on coordinate minimization [57] using the standard Lasso. The pseudocode for this algorithm is provided in Algorithm 1. A significant advantage of this method is the out-of-the-box use of the Lasso, instead of more complex and rarely implemented alternative CAP family penalties (such as the $\ell_{1,\infty}$ norm). Instead of using coordinate or gradient *descent* like others who have used dirty models, our coordinate *minimization* technique is much faster as it takes larger steps in the direction of the global optimum. To solve the objective stated in Expression 3.1, we decompose the optimization into separate problems for L and g. If

we fix L, the g that optimizes Equation 3.1 is given by:

$$\arg\min_{g} \sum_{k=1}^{K} \tfrac{1}{2}\|y^{\{k\}} - X^{\{k\}}(g + L_k)\|_2^2 + \lambda_g\|g\|_1 + \lambda_L\|L\|_{1,1} \tag{3.2}$$

$$= \arg\min_{g} \sum_{k=1}^{K} \tfrac{1}{2}\|y^{\{k\}} - X^{\{k\}}(g + L_k)\|_2^2 + \lambda_g\|g\|_1 \tag{3.3}$$

$$= \arg\min_{g} \sum_{k=1}^{K} \tfrac{1}{2}\|y^{\{k\}} - X^{\{k\}}g - X^{\{k\}}L_k\|_2^2 + \lambda_g\|g\|_1 \tag{3.4}$$

$$= \arg\min_{g} \sum_{k=1}^{K} \tfrac{1}{2}\|(y^{\{k\}} - X^{\{k\}}L_k) - X^{\{k\}}g\|_2^2 + \lambda_g\|g\|_1 \tag{3.5}$$

$$= \arg\min_{g} \tfrac{1}{2}\|\tilde{y}^* - X^*g\|_2^2 + \lambda_g\|g\|_1 \tag{3.6}$$

where X^* is the composite global design matrix in which the design matrices from each setting are concatenated vertically, and $\tilde{y}^*$ is the vertical concatenation of each $y^{\{k\}}$ adjusted for the contribution of L where $\tilde{y}^{\{K\}} = y^{\{k\}} - X^{\{k\}}L_k$:

$$X^* = \begin{bmatrix} X^{\{1\}} \\ X^{\{2\}} \\ \vdots \\ X^{\{K\}} \end{bmatrix} \tilde{y}^* = \begin{bmatrix} \tilde{y}^{\{1\}} \\ \tilde{y}^{\{2\}} \\ \vdots \\ \tilde{y}^{\{K\}} \end{bmatrix}.$$

This decomposition corresponds to the standard Lasso problem. Similarly, if we fix g, the L that optimizes Expression 3.1 is given by:

$$\arg\min_{L} \sum_{k=1}^{K} \tfrac{1}{2}\|y^{\{k\}} - X^{\{k\}}(g + L_k)\|_2^2 + \lambda_g\|g\|_1 + \lambda_L\|L\|_{1,1} \tag{3.7}$$

$$= \arg\min_{L} \sum_{k=1}^{K} \tfrac{1}{2}\|y^{\{k\}} - X^{\{k\}}(g + L_k)\|_2^2 + \lambda_L\|L\|_{1,1} \tag{3.8}$$

$$= \arg\min_{L} \sum_{k=1}^{K} \tfrac{1}{2}\|y^{\{k\}} - X^{\{k\}}g - X^{\{k\}}L_k\|_2^2 + \lambda_L\|L\|_{1,1} \tag{3.9}$$

$$= \arg\min_{L} \sum_{k=1}^{K} \tfrac{1}{2}\|(y^{\{k\}} - X^{\{k\}}g) - X^{\{k\}}L_k\|_2^2 + \lambda_L\|L\|_{1,1} \tag{3.10}$$

$$= \arg\min_{L} \sum_{k=1}^{K} \tfrac{1}{2}\|(y^{\{k\}} - X^{\{k\}}g) - X^{\{k\}}L_k\|_2^2 + \lambda_L \sum_{k=1}^{K}\|L_k\|_1 \tag{3.11}$$

$$= \arg\min_{L} \sum_{k=1}^{K} \left[\tfrac{1}{2}\|(y^{\{k\}} - X^{\{k\}}g) - X^{\{k\}}L_k\|_2^2 + \lambda_L\|L_k\|_1\right] \tag{3.12}$$

Note that each term in the sum in Expression 3.12 involves only one column of L. Therefore we can optimize each column of L independently:

$$\arg\min_{L} \tfrac{1}{2}\|(y^{\{k\}} - X^{\{k\}}g) - X^{\{k\}}L_k\|_2^2 + \lambda_L\|L_k\|_1 \tag{3.13}$$

$$= \arg\min_{L} \tfrac{1}{2}\|\check{y}^{\{k\}} - X^{\{k\}}L_k\|_2^2 + \lambda_L\|L_k\|_1 \tag{3.14}$$

where $X^{\{k\}}$ is the design matrix for setting k, L_k is the column of L for setting k and $\check{y}^{\{k\}} = y^{\{k\}} - X^{\{k\}}g$, or y adjusted for the contribution of g, where $y^{\{k\}}$ is the vector of observations for setting k. Note that we are using squared loss in all of the above equations in our development, but this is not a requirement for this paradigm. Any convex loss function would be an acceptable substitute. Pseudocode for the alternating coordinate minimization algorithm is shown in Algorithm 1.

Algorithm 1 gLOP Alternating Coordinate Minimization Algorithm

Given data $X^{\{k\}}$ (design matrix for setting k), $y^{\{k\}}$ (target vector for setting k) for settings $k = 1..K$, g_{init} (initial global model), L_{init} (initial local model)
$g_{\text{new}} = g_{\text{init}}$
$L_{\text{new}} = L_{\text{init}}$
For all $X^{\{k\}}$ concatenate into X^*
while not converged **do**
    $g_{\text{old}} = g_{\text{new}}$
    $L_{\text{old}} = L_{\text{new}}$
    for $k = \{1, ..., K\}$ **do**
        Set $\tilde{y}^{\{k\}} = y^{\{k\}} - X^{\{k\}} L_{\text{old},k}$
    end for
    For all $\tilde{y}^{\{k\}}$ concatenate into $\tilde{y}^*$
    Update g_{new} with lasso$(X^*, \tilde{y}^*, \lambda_g)$
    for $k = \{1, ..., K\}$ **do**
        Set $\check{y}^{\{k\}} = y^{\{k\}} - X^{\{k\}} g_{\text{new}}$
        Update $L_{\text{new},k}$ with lasso$(X^{\{k\}}, \check{y}^{\{k\}}, \lambda_L)$
    end for
end while

3.3 Tuning Parameters and Uniqueness

Arguably one of the most important features of the model is the penalization of L and g. To achieve this we choose the parameters λ_g and λ_L to specify which of the two should be more heavily penalized and thus likely more sparse. When viewed from a patient perspective, L can be seen to capture individual differences between patients, or if adequately sparse, patients who may be outliers in terms of physiological characteristics or other model predictors. We view g as a summary of the main effects common to all settings and want to explain as much of the variance in y as possible in this vector, reserving L for departures from the global model where necessary. However, depending on the values of λ_g and λ_L chosen, there may be no unique optimum since there may not be enough incentive for coefficients to be non-zero in one vector/matrix but not the other. In this case, many possible combinations of g and L may combine to create the optimal solution. We illustrate this in the following example: suppose $g = g^*, L = 0$ is an optimal solution. We want to use a global model only if possible. Additionally, let $L_k^* = g^* \forall k$. Setting $g = 0, L = L^*$, we get exactly the same predictions as if we set $g = g^*, L = 0$.

In order to ensure that the "global only" model is preferred, we require

$$\sum_{k=1}^{K} \tfrac{1}{2}\|y^{\{k\}} - X^{\{k\}}(g^* + 0)\|_2^2 + \lambda_g\|g^*\|_1 + \lambda_L\|0\|_{1,1} < \sum_{k=1}^{K} \tfrac{1}{2}\|y^{\{k\}} - X^{\{k\}}(0 + L_k^*)\|_2^2 + \lambda_g\|0\|_1 + \lambda_L\|L^*\|_{1,1}.$$

Because the predictions made by each side of this expression are identical, we have:

$$\lambda_g\|g^*\|_1 < \lambda_L\|L^*\|_{1,1} \tag{3.15}$$

$$\lambda_g\|g^*\|_1 < \lambda_L \sum_{k=1}^{K} \|L_k^*\|_1 \tag{3.16}$$

$$\lambda_g\|g^*\|_1 < \lambda_L \sum_{k=1}^{K} \|g^*\|_1 \tag{3.17}$$

$$\lambda_g\|g^*\|_1 < \lambda_L \cdot K\|g^*\|_1 \tag{3.18}$$

$$\lambda_g < \lambda_L \cdot K \tag{3.19}$$

Satisying this condition ensures that the global only model is preferred. However, while satisfying this condition does not prove uniqueness of the solution, *not* satisfying this condition *does* prove non-uniqueness.

3.4 Implementation and Empirical Results

Experiments were run on a server with 64 gigabytes of RAM and 16 CPU cores. The operating system used was Linux Ubuntu 12.04 (LTS). All timing and error experiments were conducted using R version 3.1.0 [43] for both gLOP and the Dirty Model, using Jalali's original implementation in R [30]). As we have shown in the previous section that gLOP can be solved using Lasso-based coordinate minimization, we were able to use out-of-the-box implementations of the Lasso. Because of this, we used the R lars package [21] to perform the individual Lasso steps used in our coordinate minimization paradigm. The folds used in cross validation were created using the R cvTools package [1].

In order to test the capabilities of gLOP on a variety of sizes of data sets we conducted four experiments with varying numbers of features and settings. The design matrices were synthesized using 0 mean and unit variance because we were not focusing on correlated features. We generated observations by adding noise (0 mean, unit variance) to $X\theta$ for each setting, where θ gives the true model coefficients for that setting. We used three

different setting-types in our experiments. Our first setting type is given by the vector of coefficients θ_1 as follows:

$$\theta_1 = \begin{bmatrix} 3 & \cdots & 3 & 0 & \cdots & 0 \\ 1 & \cdots & \frac{p}{4} & \frac{p}{4}+1 & \cdots & p \end{bmatrix}$$

Based on this, we generated two similar perturbed vectors θ_2 (less sparse) and θ_3 (more sparse):

$$\theta_2 = \begin{bmatrix} 3 & -3 & \cdots & 3 & -3 & 0 & \cdots & 0 \\ 1 & & \cdots & & \frac{p}{2} & \frac{p}{2}+1 & \cdots & p \end{bmatrix}$$

$$\theta_3 = \begin{bmatrix} -3 & 3 & \cdots & -3 & 3 & 0 & \cdots & 0 \\ 1 & & \cdots & & \frac{p}{8} & \frac{p}{8}+1 & \cdots & p \end{bmatrix}$$

For each trial, $\frac{K}{8}$ settings were generated using each of θ_2 and θ_3; the remaining $K - \frac{K}{4}$ settings were generated using θ_1. In our real-world example, the features with positive coefficients (i.e. a coefficient value of 3) would be those that increase the likelihood of a patient needing a dose of analgesia sooner than those that have no effect on the time until the next required analgesia dose (which would have a coefficient value of 0). In contrast, the features with negative coefficients (i.e. a coefficient value of -3) would be those patient attributes or physiological measurement trends that suggest that the patient will require a dose of analgesia at a later point in the future.

To choose specific values λ_g and λ_L for the following experiments, we perform cross validation but constrain the results to include only cases where $\lambda_g < \lambda_L \cdot K$. For cross validation, we first create a grid of sequences for λ_g and λ_L. We then iterate over values of λ_g, and within that, over values of λ_L to populate a grid of prediction error estimates for each combination of λ_g and λ_L. To obtain the performance for a given pair of λ_g and λ_L, we first create 10 folds. On each CV run, one of these is used as the test set, and the remaining folds are combined and used as the training set. Once convergence has been reached for the λ_g and λ_L pair, prediction error is calculated for each fold and then averaged to give the final performance which is stored in the grid. Once predictive accuracy for all pairs has been calculated, the pair of λ_g and λ_L with the lowest error is then selected from the constrained set of pairs that satisfy the criteria in Equation 3.19. When more than one pair of λ_g and λ_L attains the minimum prediction error, we select the pair with the highest value of λ_L, and then if necessary, within the set of pairs with that value of λ_L, we select the pair with the highest value of λ_g in order to obtain the simplest possible model.

Performance comparisons are shown for differences in timing and recovered data structure between gLOP and Jalali's dirty model implementation using the $\ell_{1,\infty}$ norm, the code for which is provided in R on his website [30]. To compare the runtime and error obtained by gLOP versus the dirty model, we ran 100 trials using identical synthetic data sets produced by the generative models described above for each algorithm. For each trial, the same synthetic data set with $n = 64$ was used to train both models. We tested the prediction error for each model using large ($n = 1000$) identical synthetic data test sets. λ_g and λ_L values for each experiment were obtained via the cross validation method for gLOP described above. The grid of possible combinations of the regularization parameters λ_g and λ_L used to obtain these cross validation results for each experiment is as follows:

	0	5	10	$\cdots$	100
0					
5					
10					
$\vdots$					
100					

While this may appear coarse, note that our loss function is not normalized by n, in order to compare with Jalali's implementation. Thus the range of useful λ is much wider than in some Lasso settings. In order for our results to be comparable to the results obtained by the dirty model, we used the same convergence criterion as in the dirty model implementation [30]. Specifically, we considered the algorithm to have converged if the following inequality was satisfied:

$$|(m_{\text{old}} - m_{\text{new}})| < n \cdot p \cdot K \cdot \epsilon \cdot m_{\text{old}}$$

where $\epsilon = 10^{-5}$, m_{new} is the value of the objective function calculated at the current iteration and m_{old} is the value of the objective function calculated at the previous iteration. A more stringent criterion may be desirable depending on the application of the model (i.e. to real clinical data versus synthetic data). A relatively large value of ϵ was used in our experiments due to convergence difficulties of the dirty model on data sets with large p.

3.4.1 A Simple Example

While we provide a more rigorous comparison of the two algorithms in the next section, we first ran a small experiment to provide a minimal example illustrating the differences in how

our model and the dirty model recover regression coefficients. This experiment had a small number of features and settings ($p = 4, K = 5$,$n = 64$), and used design matrices generated as described above with regression targets generated using the following underlying set of θ:

K	1	2	3	4	5
p_1	0	0	0	0	0
p_2	0	0	0	3	0
p_3	3	3	-3	0	3
p_4	3	3	3	-3	3

This data set was chosen to reflect the type of data seen when pursuing experiments with the MIMIC II data set. Specifically, we used a relatively small number of observations per setting, because few patients in the data set had more that this number of instances of analgesia administration recorded of the same type (the type of analgesia administered would change the model parameters required due to the different onset and duration of different types of opiates). Preliminary experiments and more detailed characteristics of the MIMIC II data set are provided in Appendix B. Identical data sets were used for each algorithm with $\lambda_{g/B} = 5$ and $\lambda_{L/S} = 10$. Coefficients generated by each algorithm are shown in Figure 3.2 and Figure 3.3 for gLOP and the dirty model respectively. As shown in this example, gLOP very closely recovers the θ used to generate the design matrix for each setting (two small false inclusions are present in $g + L$). In contrast, the structure recovered by the dirty model does not capture any variation between settings in S, and little variation between settings in B. In practice, having either S or B contain no non-zero values is a common problem when using the dirty model; in general it is difficult to find a pair of λ_B and λ_S such that both B and S contain non-zero values.

$$
\begin{matrix} g \\ \end{matrix}
\begin{bmatrix} 0.0000 \\ 0.0000 \\ 1.7006 \\ 2.6341 \end{bmatrix}
+
\begin{matrix} \\ \\ L \\ \\ \end{matrix}
\begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0257 \\ -0.0136 & 0.0000 & 0.0000 & 2.9871 & 0.0000 \\ 0.9240 & 1.0457 & -4.6171 & -1.4733 & 1.0145 \\ 0.1659 & 0.1166 & 0.2965 & -5.6625 & 0.2469 \end{bmatrix}
$$

$$
g + L = \Theta
$$

$$
= \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0257 \\ -0.0136 & 0.0000 & 0.0000 & 2.9871 & 0.0000 \\ 2.6245 & 2.7463 & -2.9165 & 0.2273 & 2.7150 \\ 2.8000 & 2.7507 & 2.9306 & -3.0283 & 2.8810 \end{bmatrix}
$$

Figure 3.2: g and L coefficients generated by gLOP for a minimal example

$$
\begin{matrix} \\ \\ B \\ \\ \end{matrix}
\begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 2.4858 & 2.4858 & -2.4858 & -0.5812 & 2.4858 \\ 2.6487 & 2.6487 & 2.6487 & -2.6487 & 2.6487 \end{bmatrix}
+
\begin{matrix} \\ \\ S \\ \\ \end{matrix}
\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
B + S = \Theta
$$

$$
= \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 2.4858 & 2.4858 & -2.4858 & -0.5812 & 2.4858 \\ 2.6487 & 2.6487 & 2.6487 & -2.6487 & 2.6487 \end{bmatrix}
$$

Figure 3.3: B and S coefficients generated by the dirty model for a minimal example

3.4.2 Scalability and Accuracy

To examine and compare the scalability and accuracy of gLOP and the dirty model on different sizes of data sets, four timing experiments were conducted for different sizes of data sets: small p, small K; small p, large K; large p, small K; and large p, large K. For each experiment, 100 trials were run using data generated as described above. Identical penalty coefficients were used for both algorithms; these were generated from the gLOP cross validation. Timing results for both algorithms are shown in Table 3.1. As hypothesized, gLOP performs significantly faster than the dirty model on all data set sizes; however, this effect is particularly noticeable when p is large. Similarly, the model error for gLOP is significantly lower than the error for the dirty model; again, this is shown most clearly when p is large. MSE was computed for each trial using a separate test set of size $n = 1000$. The means and standard deviations of the timing and MSE of each experiment are shown in Table 3.2. Independent two-sample t-tests were used to determine statistically significant differences for run-time and model error between gLOP and the dirty model. Significant differences ($p < 0.05$) between gLOP and the dirty model are indicated by a *.

p	K	n	gLOP	Dirty Model
16	16	40	0.1822s	0.2873s*
			±0.0117	±0.0212
16	128	40	0.8122s	2.2379s*
			±0.0296	±0.0672
128	16	40	1.1648s	16.3024s*
			±0.0444	±0.4156
128	128	40	8.9316s	118.447s*
			±0.1481	±3.5134

Table 3.1: Run-time results for gLOP versus the dirty model

3.5 Algorithm Complexity

As shown in the previous section, gLOP performed significantly faster than the dirty model on all data set sizes tested. In order to explore possible reasons behind this result, we analyzed the computational time complexity of gLOP, LARS and the dirty model. Jalali's implementation of the dirty model [30] was found to have a computational time complexity

p	K	n	gLOP	Dirty Model
16	16	40	1.3931	6.3718*
			$\pm$0.0637	$\pm$0.2599
16	128	40	1.4602	2.2171*
			$\pm$0.0237	$\pm$0.121
128	16	40	93.6959	141.1617*
			$\pm$7.5097	$\pm$9.5854
128	128	40	73.9881	141.1624*
			$\pm$2.7155	$\pm$3.7506

Table 3.2: Test error results for gLOP versus the dirty model

of

$$\mathcal{O}(p^2Kn + (p^2K + pKn) \cdot i),$$

where i is the number of iterations performed by the algorithm before convergence. In contrast, the time complexity of LARS, the subroutine used in gLOP's coordinate minimization paradigm, is

$$\mathcal{O}((A^3 + An) \cdot j + p^2n),$$

where A is the size of the active set and j is the number of iterations performed by the LARS algorithm before convergence. Using this result, the time complexity of gLOP was found to be:

$$\begin{aligned} &\mathcal{O}((npK + \text{LARS}(nK, p) + K \cdot \text{LARS}(n, p)) \cdot i) \\ =&\mathcal{O}(\big[pKn + p^2n(K+1) + \big[AKn + A^3(K+1)\big] \cdot j\big] \cdot i). \end{aligned}$$

We hypothesize that the large gains in performance seen in gLOP versus the dirty model are due to fact that gLOP's performance is bounded by the size of the active set, whereas the dirty model complexity is bounded by the number of features, p. In the worst case when $A = p$, gLOP's complexity is actually worse than the dirty model, since that factor is cubed in gLOP instead of squared as in the dirty model. However, we expect that this will seldom happen in practice, as we assume a moderate to high level of sparsity in the types of data intended for use with this model. In fact, we know that $A \leq \min(n, p)$ [12].

Chapter 4

Future Directions

As shown in the previous section, gLOP outperforms the dirty model on timing and thus scalability, and also provides an interpretable model that captures both global effects and individual setting-specific effects. Because our algorithm was designed with subroutines already implemented on big data systems (such as Hadoop [6] and Spark [60]), we expect that gLOP will be suitable for problems using big data. However, further work remains to be done to improve and test the model for future use on a large real-world data set.

4.1 Correlated Features

In the domain of predictive pain management, often features are highly correlated which is problematic for models such as the Lasso, because if the true coefficients of correlated features in different settings have opposite signs in each setting, the resulting effects tend to cancel with each other, giving poor results or results where no features have non-zero coefficients. We hypothesize that this will not be a problem for gLOP because while variance in y may not be well explained in the global model g, these effects will be captured in L. In this case, L would not be as sparse as if features were not correlated with opposite coefficient signs in the underlying distribution, but gLOP will still provide an interpretable model. In order to confirm this hypothesis we intend to run experiments using correlated features generated using the methods described in the previous section to compare the recovered structure of the data to the true model coefficients, and to the structure recovered by the dirty model. Following this additional testing, a further avenue of future work is the development of an R package for small-scale to medium-scale data, and the development of a Hadoop or Spark package for working with larger data.

4.2 A Single-Lasso View of gLOP

A less efficient but equivalent way of structuring the data is to create a large $n \cdot K \times p \cdot (K+1)$ block matrix with the first p columns corresponding to the composite design matrix of all settings, horizontally concatenated with a block matrix with design matrices for each setting on the diagonal:

$$X^* = \begin{bmatrix} X^{\{1\}} & X^{\{1\}} & 0 & \cdots & 0 \\ X^{\{2\}} & 0 & X^{\{2\}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X^{\{K\}} & 0 & 0 & \cdots & X^{\{K\}} \end{bmatrix}.$$

Using this structure, we are able to use a standard single Lasso to optimize the following expression:

$$\min_{\beta} \|y^* - X^*\beta\|_2^2 + \sum_{i=1}^{p \cdot (K+1)} \lambda_i |\beta_i|$$

where y^* and β are defined as:

$$y^* = \begin{bmatrix} y^{\{1\}} \\ y^{\{2\}} \\ \vdots \\ y^{\{K\}} \end{bmatrix} \beta = \begin{bmatrix} g \\ L^{\{1\}} \\ L^{\{2\}} \\ \vdots \\ L^{\{K\}} \end{bmatrix}.$$

While this representation of gLOP is not space efficient, the structure gives us the ability to use properties of the Lasso to formally establish asymptotics in n and/or K and p. We have provided basic conditions detailing when the gLOP solution will not be unique, depending on values chosen for λ_g and λ_L. However, much work remains to be done in establishing more thorough formal conditions for uniqueness of the solutions generated by gLOP. While developing conditions that guarantee uniqueness for our algorithm will be a nontrivial amount of work, the use of the Lasso over iCAP gives us an advantage in the types of constraints we must impose on our design matrices. Using the condition specified in Equation 3.19, if we fix $\lambda_g = c \cdot \lambda_L$, we intend to explore the creation of a LARS-like algorithm capable of computing the full regularization path for gLOP at a fixed c [12].

Another future direction in more theoretical work remains in the formalization of "prediction outliers", or entities that are "outliers" in the sense that they require different

predictive models from the majority of settings or individuals. It is possible that we may be able to match new patients with existing global models trained using previously collected patient data. If we are able to obtain high quality big data, we will be able to make predictions on whether the global model will be suitable for a new patient, whether that patient will require a new local predictive model, or whether a new global model should be trained using that patient's data.

4.3 Application

Finally, we intend to test this algorithm on a real-world data set for predictive pain management. In the development of this thesis we investigated the prospect of using data collected from the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-II) project [36, 19] (see Appendix B for full details of the data and preliminary analyses performed). This data set was collected at the Boston's Beth Israel Deaconess Medical Center (BIDMC) beginning in 2001 by a collaborative team with MIT and Phillips Medical systems, and currently contains data from approximately 32,000 patients. While both waveforms and clinical data are recorded in the data set, several challenges rendered us unable to use the data set for the current work. The chart data was very sparse with few (varied) measurements in each time epoch. In general, measurements had varying amounts of time between them, dependent on when the patient was last checked by the nurse or clinician on duty. Many chart records were incomplete (for example, including timing of analgesia administrations, but missing the type of analgesia administered at each time point). Additionally, analysis was complicated by the labeling in the data set. Recording acronyms and format differed between patients, making automated data extraction much more difficult (this included misspelled words and acronyms). Finally, while many analgesia administration events were labeled as being patient-controlled, the MIMIC-II data is generally derived from patients in the ICU. In this case, it is possible that such PCA events were actually administrated by nurses, or automatically by a PCA unit attached to an unconscious patient.

The data cleaning necessary to render the data suitable for use with our paradigm made the use of MIMIC-II unusable for the purposes of this thesis. Additionally, few patients in the data set matched our criteria (patients on fentanyl administered via a PCA unit). For this reason, it is possible that a new data set better suited to our analyses would need to be collected in order to test the method for use in the context of predictive pain management. If data were gathered directly from PCA machines, the problem of incomplete dosage administration would be ameliorated. Once we have an appropriate data set we will test

the method with the data as a whole, then simulate how this model would work in the context of real-time prediction of analgesia requirements. For the real application of this model to clinical decision support problems, we propose training the model on already-acquired data from several patients. Once the model is trained, we will use the global model as a starting point, then refine it as we accumulate more patient data. A related avenue of future work regards the interpretability of the model for physicians, and feasibility of implementation. Conducting focus groups or a panel discussion with physicians on the best output format to increase the interpretability of gLOP from a clinical standpoint and barriers to adopting this type of model in practice would be valuable for the future goal of building a real-time system for predictive pain management.

Chapter 5

Conclusions

In this work, we addressed the issue of developing a new multitask learning paradigm suitable for use in predictive pain management, and in healthcare applications in general. Challenges for developing such a method included the need for interpretability; a dirty model, as we must model both global population-wide effects and individual effects; and the need for scalability, as big data is common in many healthcare applications. In particular, we desired a model suitable for data from many patients (large K) but not large amounts of data about each individual patient (small n). To accomplish this task, we aimed to address the following research questions:

- How can we adapt and design an MTL model suitable for healthcare contexts that gives us a more interpretable predictive model than existing models?
- How can we design such a model to scale to large data better than the existing dirty model?
- How do the recovered coefficients differ between our model and the dirty model, and does this affect interpretability?

We addressed these during our development of gLOP, which lead to the following contributions:

1. We proposed a new "cleaner" dirty model, gLOP, for multitask learning.

2. We gave an algorithm to solve this problem that uses coordinate minimization and existing implementations of the standard Lasso to reach the global optimum. We demonstrated that this coordinate minimization technique provides clear computational advantages over the standard dirty model.

3. We demonstrated that the interpretability of gLOP is superior to the dirty model, while still accurately representing the underlying structure of the data.

In this thesis, we presented a novel statistical method using predictive models as a contribution towards a full solution to the problem of PPM. While predictive models have been extensively studied in the field of machine, this is a novel application setting. Our model can be used in many applications appropriate for multitask learning and dirty models in particular (such as classification of handwritten digits written by many different individuals [31]). Specifically, we observed that by combining the underlying principles behind the interpretation of ANOVA results and the structure of dirty models, we were able to obtain an interpretable model capable of prediction that can accurately recover underlying data structure. A significant contribution of this thesis is a novel machine learning tool applicable to many different problem domains.

An additional advantage of our alternating Lasso coordinate minimization algorithm is that it uses subroutines already deployed on off-the-shelf systems for big data analysis. We plan to release an R package for small-scale and medium-scale data, and a Hadoop or Spark package for working with big data. Further future work includes establishing asymptotics and further conditions for uniqueness for our algorithm, formalizing the concept of "prediction outliers" and testing our method with a real-world clinical data set suitable for predictive pain management.

APPENDICES

Appendix A

Predictive Pain Management Background

Many attempts have been made to quantify pain for the purposes of improving patient care in postoperative settings. However, this is a challenging problem because there has been no consensus in the medical or research communities on how to directly observe pain without proxy measures or patient self-report. Surrogate or proxy features that have been used to quantify pain or predict analgesia consumption include demographic and patient features (such as characteristics of surgical settings and patient demographics), quantitative physiological and treatment-based measurements, and behavioural indicators of pain (such as facial expression and other movements). Developing objective measures to quantify behavioural and other indicators of pain would be an asset in establishing and maintaining a treatment regime for pain, given the current controversy over which indicators are most indicative of when a patient is in pain. While some work has been done to this end in the field of computer science (and machine learning in particular), including using machine vision to detect facial expressions indicative of pain [53, 4], many studies about indicators of pain have not been done in a temporal setting where we wish to predict analgesia consumption at some point in the future, and are therefore potentially unsuited for the prediction of pain at specific points in time, given a sequence of patient data. Still other studies have examined the use of many different machine learning methods (such as decision trees, support vector machines (SVM), etc.) to predict the need for pain consultations and PCA readjustment [52] and the need for femoral nerve block following surgery [51], but experimental bias due to a lack of rigor in the evaluation process of these methods has left the issue unclear as to which techniques are most appropriate for analyzing patterns in medical data that are indicative of the need for analgesia administration.

Goals for Predictive Modelling

The model we propose to develop for the eventual purpose of clinical decision support in pain management will be predictive and use time-varying inputs. Predictive models in this context are those that seek to predict future pain status, or future analgesia requirements. Non-predictive models are those designed to assess current pain status or analgesia requirements. Time varying inputs are features whose measurements change over time, such as physiological waveforms. Predictions or assessments may be made at many points in time using time-varying inputs. In contrast, non-time-varying inputs are measurements of unchanging physiological characteristics, demographics, or other clinical information at a single point in time. Generally only one assessment or prediction at a single point in time is made using these features. Definitions of these model feature characteristics are summarized in Table A.1. As a reasonable first step, feature selection will be performed for this problem using gLOP, the method developed in this thesis, and following this, a predictive model for pain management will be tested once appropriate data is acquired.

In this appendix literature examining methods of detecting pain and predicting analgesia consumption will be discussed with regard to these definitions and critiqued.

Predictive Model	Goal: predict future pain status analgesia requirements
Non-Predictive Model	Goal: assess current pain status or analgesia requirements
Time-Varying Inputs	Features whose measurements change over time (eg. physiological waveforms) Predictions/assessments made at multiple points in time
Non-Time-Varying Inputs	Features whose measurements do not change over time (eg. demographics) Prediction/assessment made at a single point in time

Table A.1: Definitions of PPM model and feature types

A.1 Inferring Current Pain Status

The following work has explored inferring current pain status. In general, these studies have not attempted to predict future pain, particularly for the purposes of developing a model to control analgesia administration.

A.1.1 Behavioural Inventories and Physiological Assessment

While quantitative methods of assessing pain using physiological indicators are limited (i.e. methods that do not rely on subjective assessments of pain by a third party), researchers and clinicians in the field of pain management have previously attempted to derive measures that could be used in real-time by care providers to estimate the amount of pain experienced by patients. An example of this type of measure is the Critical-care Pain Observation Tool, which is an inventory that identifies behavioural indicators of pain [17]. Specific items on the CPOT include facial expression, body movements, muscle tension, and ventilator compliance or vocalization, depending on intubation status. Within these categories, the specific behaviours that are indicative of pain (the behaviours with the highest pain rating) are described as grimacing, restlessness, muscle tension and either fighting the ventilator or crying or sobbing. Each of these high level descriptions in the scale is accompanied by a detailed low-level description to help with scoring. The assessment of the validity of this tool relied on patients to indicate whether or not they were in pain by nodding their heads at intervals following the surgery; nurses used the CPOT to assess the patients behavioural pain indicators concurrently, and their answers were compared with the patients' self reports. CPOT scores increased when patients were engaged in an unpleasant procedure (such as being moved, etc), and CPOT scores differed significantly from when patients indicated that they were in pain, versus when they were not in pain. Additionally, unconscious patients had higher CPOT scores when experiencing a nociceptive procedure, indicating that pain is likely experienced despite a lack of consciousness. However, some limitations were evident in this study; only two nurses were involved in rating patient pain, limiting the generalizability of the findings. Additionally, patients were only asked to give a dichotomous response as to whether they were feeling pain or not. This is reasonable given the condition the patients were in, but it does not allow a comparison of how much relative pain a patient would be feeling, and how their score (for example, on a visual analog scale) would compare to the score obtained on the CPOT. Such information could be very useful for predicting the amount of analgesia that might be required for a patient at a given time. Another inventory for behavioural indicators of pain is the Behavioural Pain Scale (BPS) [42], which uses indicators very similar to the CPOT. Relevant behaviours listed on the BPS are facial expression (grimacing), upper limb movement (permanent retraction), and compliance with ventilation (inability to control ventilation).

Pain assessment techniques for specific sub-populations of patients have also been addressed in the literature. Neonatal patients are an example of such a population, as they are unable to deliberately communicate with the care team to express pain, and instru-

ments designed for assessing pain in adults may not be appropriate for this age group [2]. A survey of the literature by Altenhein (1998) [2] found that decreases in transcutaneous oxygen levels, increases in heart rate and respiration, vagal tone changes, behavioural changes (such as variations in crying), palmar sweating, and increases in blood pressure, facial expression, and movement are some factors that may be indicative of pain experienced by neonatal patients. More research has argued that many of the factors previously mentioned are not specific to the experience of pain: for example, crying may be a sign of hunger or many other conditions [44]. In this paper, age, whether or not the patient was sleeping, and previous experience of pain were some of the many factors discussed that may affect pain response in neonates.

While this information is useful for building a future predictive model for pain management, the studies described did not consider how these physiological effects might change in response to increases in pain over time, or in response to the administration of opioids at a future point in time. Because of this, the results of these studies may not be directly applicable to our prediction problem, but provide a useful guide for features to explore when performing feature selection for this application.

A.1.2 Facial Expression Monitoring

Many behavioural inventories include facial expression as an indicator of the level of pain a patient is currently experiencing [42], and other researchers have also found facial expression to be an indicator of pain in multiple patient populations [3, 35, 4]. The problem of automatically monitoring facial expression for this purpose has also been studied by researchers in the field of machine learning, and machine vision in particular. A relatively recent but limited study proposed an image acquisition system to detect discomfort in elderly patients with cognitive impairment, however, this system did not incorporate prediction techniques to predict pain [5]. Instead, labeling software was provided to experts who could label when a behavioural indicator of pain had occurred in the film, according to a behavioural pain inventory. A more advanced study used machine vision to track head movement and identify facial expression as indicators of patient agitation [4]. This was done in a real-time manner, to be of practical use in a clinical setting. The algorithm used in this study had five main steps: detecting the patient's head position, putting a boundary around the face in the image, segmenting the face, evaluating grimacing, and finally, computing patient agitation based on facial expression in combination with other measures known to be indicative of agitation (such as heart rate and blood pressure). The main measure of grimacing used by the authors was the presence of wrinkling in the face; edges representing these wrinkles were extracted from the image. Before this was

done, high-pass filtering was performed in areas where wrinkles would be more likely to appear. Areas where wrinkles are unlikely to appear were ignored, to facilitate dynamic identification of grimacing.

A limitation of this study was described by Gholami et al. (2010): the authors did not consider other facial expressions that may create facial wrinkling [18]. In contrast, Gholami et al. framed pain detection in neonates as a classification problem, and implemented a sparse kernel machine (relevance vector machine) to classify images of infants as being in pain, or not in pain, based on the framework of support vector machines (SVMs). The inputs to the algorithm were a set of target values (Z), and a set of input values (X). This probabilistic classification model works by learning a latent function $y(x)$ using regularized kernel regression. The *predictive distribution* predicted what target value $z \in \{0, 1\}$ was associated with a new input $x \in \mathbb{R}^D$ (predictive distribution: $p(Z = z|X = x; X, Z)$). The probability of a new input x belonging to a specific class C_1: $p(C_1|x, X, Z)$ was obtained by putting $y(x)$ through a logistic sigmoid function $\sigma\big(y(x; X, Z)\big)$. The authors used variables from the infant classification of pain expression (COPE) database to test and train the classifier. The database contained 204 faces of infants when exposed to various stimuli. The reaction to three of these classes of stimuli were labelled as non-pain, including crib transport, air (a puff of air on the infant's nose), friction (having the heel rubbed with a piece of cotton soaked in alcohol). The last stimulus, extracting blood from the infant's heel by puncturing it, produced infant expressions labeled as pain. The algorithm was trained on various faces from the pain and non-pain classes, and produced a posterior distribution of the difficulty of categorizing the pain status of the infant in various photographs. The posterior distribution could be thought of as corresponding to the intensity of pain experienced by the infant; this observation was validated by comparing the results of the algorithm with human (expert and non-expert) assessments of the infant's pain intensity. The authors also concluded that this method could be of additional use in clinical decision support systems for sedation and analgesia in the ICU.

While Gholami et al.'s method of introducing a posterior probability of the difficulty of categorizing facial expressions produced better classification results than more simplistic machine learning algorithms previously tested, looking at singular facial expressions in isolation for the classification of pain versus non-pain has been discussed as a limitation of this method of pain identification in general. An earlier study attempted to address this problem by introducing a temporal component to facial recognition, as temporal dynamics in facial expression provide rich source of information for interpreting human behaviour [53]. Instead of using a binary pain versus non-pain classification scheme, Valstar et al. used a multiclass one-versus-one SVM on frames of a video to identify whether a face was in one of four different temporal phases: the onset phase (where the facial expression

begins to change), the apex (where the facial expression is at its peak), the offset phase (where the facial muscles are relaxing) and the neutral phase (where there is no trace of the previous facial expression). Timing dynamics were incorporated into the classification model by using a Hidden Markov Model (HMM) in combination with the SVM, a method which significantly improved the classification of facial expressions. Specifically, the model identified and tracked a set P of 20 key facial points using Particle Filtering with Factorized Likelihoods (PFFL). Before running the PFFL algorithm, feature vectors were calculated for each pixel in each region of interest (ROI) corresponding to a key facial point based on the grey values of the patch (13x13) surrounding the pixel and responses to Gabor filters (based on orientation and spacial frequency) to establish a template of the facial point. To identify how much the facial expression had changed, the x and y coordinate deviations between the point in the current frame $p_i \in P$ where $i = [1 : 20]$ and each frame j were calculated as the first two features for each facial point. Additionally, for each pair of points, the distance between them, the difference in the distance between them relative to the distance between them in the first frame, the angle made by connecting the two points to the horizontal access and finally, the first temporal derivative of each feature were calculated. To combine the SVM and HMM, the probability of the output of the SVM on an input feature vector x, $f(x)$, was estimated by fitting the input vector with a sigmoid function. The HMM had a state for each phase of facial expression; the outputs of the SVM, pairwise probabilities of whether a feature vector x belongs to class q_i (the first member of the pair) or q_j (the second member of the pair), were first transformed to posterior probabilities $p(q_i|x)$ and finally emission probabilities $p(x|q)$ by Bayes' rule [53]. This method resulted in higher classification accuracy than using an SVM alone.

Based on the results found from the study of facial expressions in the context of pain, detecting the facial expression of patients in postoperative settings to predict pain would likely improve the accuracy of the proposed model. However, capturing video and images of patients in a hospital setting would pose an ethical dilemma. Practically, the effort required to process the data and extract features would limit the ability of the model to compute predictions of pain in real-time.

A.1.3 Pain Status Inference Summary

The work reviewed on inference of current pain status may be informative for initially testing gLOP with clinical data, as the previous work and gLOP are both non-predictive in the sense that they do not seek to predict when a future analgesia event will occur, and as they both use features derived from patient monitoring. However, the features used in the behavioural inventories and physiological assessments are not time-varying. While one of

the facial expression monitoring studies did use time-varying features, ethical limitations of filming patients during recovery on a wide scale render these particular features unfeasible for the current work. The broad goal of the work on inferring current pain status does not align well with our eventual goal of predicting analgesia requirements, since that task requires inferring pain status at some future time. Additionally, these studies do not explore features relevant to control of analgesia administration.

A.2 Predicting Treatment Demand for Pain Management

Although efforts have been made in the medical community to understand factors affecting the amount of analgesia needed by patients to mitigate pain, there is mixed evidence on how patient factors such as demographic information, characteristics of hospital setting and surgical type, and physiological measures may inform clinicians and researchers about the amount of analgesia a patient will need during recovery, or future requirements for other pain management strategies. Many studies have explored methods of predicting the demand for future pain management treatments and analgesia requirements.

A.2.1 Regression Techniques for Predicting Analgesia Consumption and Reported Pain

In an early study, Chia and colleagues [10] found gender to be a significant predictor of analgesia consumption; specifically, it was found that females consume less PCA than males following surgery. In this study, age, education, body weight, and education sites were found to be insignificant factors. However, a more recently conducted literature review found gender to be insignificant as a predictor of PCA consumption [29]. Additionally, age, type of surgery, preoperative pain and anxiety were found to be significant predictors of PCA usage in this review, in contrast to the previously described work. In another study on factors related to anaglesia requirements following surgery, Ekstein and colleagues [14] found that patients undergoing orthopedic pain were more likely to suffer from severe pain more frequently than patients undergoing laparotomy. Severe pain in orthopedic patients was found not to be mitigated successfully by morphine; the patients required a more concentrated dose of morphine than would be typical, supplemented with ketamine, in order to experience relief from pain.

Factors have also been explored for the prediction of pain and analgesia consumption in patients undergoing elective cesarean sections [40]. In addition to routine demographic and patient information, the patients completed the State Trait and Anxiety Inventory (STAI), reported preoperative pain scores and expectation of pain following surgery using the Visual Analogue Scale (VAS), and were measured for threshold and sensitivity to thermal pain, suprathreshold thermal pain intensity and unpleasantness, and thermal pain threshold temperature, both on the forearm and back where relevant. The authors performed factor analysis [24] on the data collected, which resulted in six main predictive factors: pain and unpleasantness in response to thermal stimuli, preoperative blood pressure (BP), preexisting pain, pain expectation, thermal pain threshold, and intraoperative factor (duration of surgery and sensory blockade level). These were used to predict a variety of outcomes using multiple regression. Resting pain was best predicted by pain and unpleasantness resulting from the administration of noxious thermal stimulation, and pain expectation. Back thermal pain threshold was the best predictor of evoked pain, while pain and unpleasantness and BP were the best predictors of the composite pain score. Preexisting pain was the single best predictor for intraoperative analgesic requirement, whereas analgesic requirement in the recovery room was best predicted by thermal pain threshold and STAI; STAI alone was the best predictive model for total analgesic requirement [40].

A.2.2 Classification for Predicting Pain Treatment Adjustment

A later study performed by Tighe et al. [51] contrasted various machine learning techniques to predict the need for femoral nerve block (FNB) in combination with more conventional methods of analgesia, following anterior cruciate ligament (ACL) repair. This problem was framed as a binary classification problem (whether or not a FNB should be placed in the patient), and used only preoperative patient information including demographic information such as age and gender, BMI, preoperative pain, drug and alcohol use, age, gender, and analgesics and anxiolytics used before the surgery. Surgical factors were also used for prediction, such as the graft type, surgical approach, and type of anaesthesia. Data was analyzed for 349 patients. In total, 20 factors were used for predicting whether patients would require FNB; an alpha level of 0.05 was employed for this analysis. The authors hypothesized that the performance of machine learning classifiers to predict whether patients would require FNB would be comparable to the performance of logistic regression on this data set.

The machine learning methods used in this study were standard WEKA implementations of logistic regression, BayesNet, multilayer perceptron, SVM and alternating decision trees (ADTrees). The boosted logistic regression used a heuristic stop at 50 iterations, a

maximum of 500 boosting iterations and did not assign error to probabilities. The BayesNet classifier used the K2 algorithm that attempted to add parent nodes to each node in the directed acyclic graph, and incorporated a Markov blanket classifier that facilitated grouping parent and child nodes. The multilayer perceptron was built with two hidden layers, and the ADTree underwent 10 boosting iterations. Classification was performed without optimizing the classifiers for the particular problem at hand; default configurations for each classifier implementation were used. The classifiers overfit the data (the SVM was particularly bad for this), perhaps because of the small sample size used for training and the lack of parameterization of the model for this specific problem. Despite this, the ML classifiers were found to perform as well as, or better than, logistic regression in terms of the area under the receiver operating curve. However, logistic regression outperformed all of the classifiers for percent correctly classified. Factors that were related to the need for FNB included gender, tobacco use, and types of analgesia used during surgery. The authors concluded that using ML classifiers led to improved classification ability for whether patients will require FNB.

While Tighe et al. concluded that more complicated machine learning techniques can improve classification ability in the context of medical data despite similar classification accuracy to logistic regression (the standard machine learning technique), several aspects of the experimental protocol employed by the authors suggest that the results obtained may not accurately reflect the potential performance of the algorithms. Because the classifiers were not correctly parameterized (most settings used were the default for the algorithms in WEKA), the potential performance of the classifiers cannot be accurately obtained by this type of experiment. This limitation was also cited by the authors as a subject for future research. Additionally, using k-fold cross validation to test many classifiers without withholding a validation set may cause the results of such experiments to appear more accurate than the actual performance of the classifiers. If a validation set is not withheld until the end of the experiment for testing, the classifiers together would have access to the full data set, which would result in biased estimates.

In order to address the issue of the lack of parameter optimization in this study, the authors subsequently conducted a study employing similar methods to attempt to predict whether patients would require preoperative acute pain service (APS) consultations [52]. In this study, the authors hypothesized that predictive analytics could be employed to accurately predict whether patients would require APS consultations before a request was made, using information readily available in electronic medical records. A variety of classifiers were used in this retrospective cohort study, including Bayesian (BayesNet, Naive Bayes), function based (logistic regression, SVM, multilayer perceptron, radial basis frequency network, voted perceptron), lazy (K-nearest neighbour), rule based (decision

trees, propositional rule learner, PART decision list, ZeroR) and decision tree based (J48, ADTree, Random Forest) classifiers. The SVM, logistic regression and multilayer perceptron were excluded from the study due to the amount of processing time required to partially train the classifiers.

In order to find the best possible classifier and optimize performance, the authors developed ensembles and attempted to reduce the dimensionality of the data. For the first method of optimization, the authors constructed a meta-classifier using the models with the highest area under the receiver operating curve (AUC) scores, which gave a single prediction from the classifiers using a vote system which averaged the probability estimates of the individual classifiers. The dimensionality of the data set was reduced using correlation-based feature subset selection (CFS), which chooses features that are not very well correlated with other features, but are correlated highly with the separating class. Of 11 total factors, attributes that were found to be related to whether or not APS consultations would be required were the anaesthesiologist, surgeon, primary and secondary current procedural terminology codes, and the start time and location of the operating room. The classifiers were evaluated on area under the receiver curve (AUC), accuracy, sensitivity, specificity and CPU time required. The classifiers had varying levels of performance (average accuracy 92.3%; differences significant at $\alpha > 0.05$), with ZeroR exhibiting the lowest (baseline) classification accuracy (84.2%), and the Voted Perceptron having the highest level of accuracy (94.3%). The sensitivity and specificity also varied widely among classifiers, with BayesNet and NaiveBayesUpdateable tied for the highest specificity at 87%. When the authors created an ensemble of the classifiers to optimize performance, no significant difference in performance (measured by AUC; $\alpha < 0.05$) was observed between the metaclassifier and the individual classifiers. Similarly, no improvement in classification performance was observed as a result of optimizing by reducing the dimensionality of the data set, although the time required for training the models was reduced [52]. Based on these results, the authors concluded that using ML classifiers to assist with the prediction of which patients will require an APS consultation is feasible and could be very useful to physicians in terms of monetary and economic resource expenditure. Additionally, the authors concluded that the computational time requirements for this process can be reduced by reducing the dimensionality of the data set. This is one of the advantages of gLOP, our newly introduced model. Because gLOP performs feature selection and its computational complexity is based on the size of the active set instead of the number of features, in practise our method is capable of much faster performance than existing models for similar applications (i.e. the dirty model).

Some of the limitations cited by the authors of this study included a lack of specific patient variables (such as physiological patient parameters, and patient preferences elicited

from discussions about treatment options) for use in training the classifiers. The exclusion of this information (due to lack of availability in the patient records used) may have biased the classifiers towards lower performance than would be observed if the authors had access to this relevant information [52]. Another methodological limitation of this study is the lack of hypothesis driven experimentation regarding which classifiers would be the best fit for the data. As in the authors' previous experiment, experimental bias was greatly increased by running multiple tests on the same data and then forming conclusions about classifiers based on combined performance.

A.2.3 Decision Trees for Predicting Analgesia Demands

In contrast, Hu et al. [28] more recently used decision trees to categorize post-operative PCA (within 72 hours of the surgery) for 1099 patients into three symbolic levels of consumption (low, medium and high), as opposed to the more popular method of regressing on predictive variables to produce a continuous numeric target variable. The authors used four categories of predictors to train the classifier: demographic factors (such as age, gender and weight); biomedical factors (such as pulse, systolic and diastolic blood pressure, whether the patient had diabetes, hypertension and/or acute myocardial infarction, and preoperative patient health status or disease severity); operation related factors (such as the class of operation, the duration of surgery, urgency of procedure, and type of anaesthesia used); and various factors related to PCA dosage and timing.

Because the predictive accuracy of decision trees may be increased by using an ensemble of trees instead of a single tree, the authors used bagging to create different versions of predictors, which were then aggregated to form a single predictor. A stratified k-fold cross validation experiment was performed to evaluate the classification accuracy of the C4.5 decision tree algorithm, as compared to a variety of other classification methods including the C4.5 with bagging, C4.5 with the AdaBoost ensemble algorithm, Artificial Neural Network (ANN), SVM, Random Forest, Rotational Forest, and Naive Bayesian classifiers. The bagged C4.5 algorithm had significantly higher overall accuracy (which combined sensitivity and precision for the low, medium and high categories) than any of the other methods tested, even after correcting for multiple comparisons between other classification methods.

This study also aimed to predict whether patients would require their PCA settings (specifying the allowed dosage rate) to be readjusted within 48 hours following surgery. There was imbalance in the class ratio between patients requiring readjustment (19%) and patients not requiring PCA readjustment (81%). Because of the imbalance in the data

set, overall accuracy was not used as a performance measure, as the performance of the classifier on the negative class would bias the overall accuracy of the classifier; instead sensitivity, precision, specificity and F-score were used as performance metrics. Because the decision tree classification method had the highest overall accuracy for predicting total PCA usage, decision trees with bagging and boosting were tested on the imbalanced data set, and were subsequently tested with biased sampling (over-sampling of positive cases, and under-sampling of negative cases) to balance the class ratio. The Random Forest method was also compared to these methods for PCA readjustment. The bagged decision tree with under-sampling of the negative class led to the best F-score for classification performance.

As well as having higher predictive performance for both total PCA usage and PCA readjustment requirements, the authors argued for using decision trees (instead of more complex statistical methods) due to their comprehensibility for medical staff. Because the decision tree algorithm is easily visualized and can be translated into a set of if-then rules the authors argued that healthcare delivery staff would be more likely to understand how PCA was being predicted given a graphical representation of the tree, and thus would be a more useful tool in a clinical context than more opaque techniques. However, because the authors used bagging to create the most accurate decision tree for classification, this justification may not be applicable, as the decision process is more complicated than a simple series of if/then statements. It is possible that probabilistic temporal models could be explained with a similar level of ease to physicians (given that the authors were not just using a basic single decision tree), and would have the additional advantage of being able to predict PCA analgesia requirements in real-time. This is an avenue of future work that we intend to explore regarding the presentation of results from our method gLOP. Clinician and caregiver input about the best ways to communicate the information obtained using our statistical model will be invaluable for eventually implementing a model for use in clinical settings.

In comparison to the previously described work on inferring current pain status, the goals of these studies predicting treatment demand for pain management are more aligned with those of the future application of gLOP. However, these models do not use time-varying features, and largely do not use features derived from patient-monitoring. Additionally, many features found to be predictive in the last study described in this section were related to characteristics of analgesia consumption. As we would like to predict outcomes of this type in future work, these features are not directly related to the problem of predictive pain management. However, like previously described work, the results from these studies may provide a useful basis for considering feature selection if we are able to access high quality chart information as well as inputs from physiological measurements

for testing gLOP in the context of predictive pain management.

A.3 Control Models for Regulating Analgesia Administration

Finally, in a study most related to the proposed application of gLOP Hu et al. [27] proposed a partially observable Markov decision process (POMDP) to model and control the concentration of anaesthesia in a patient's bloodstream during surgery, as both overdosing and underdosing may carry severe consequences for patient health.

The authors used a biological compartment model for the patient, where the body is modelled as a series of connected finite compartments that transfer fluids at a linear rate between each other and the outside world. The goal in this problem would be to bring the patient to the proper drug concentration as quickly as possible, and then maintain that rate through additional infusions over time as the drug is cleared from the system. The patient's pharmacokinetic parameters of compartment volume (v; litres), drug clearance rate (c; litres/time), and drug infusion rate ($a(t)$; g/litre) were used to model the concentration of the drug in the patient's system, and constituted the state space for the problem. With the assumption that meeting the target blood concentration level is equivalent to the best treatment possible, the problem was initially framed as a dynamic programming program, and then subsequently as an infinite horizon POMDP. It was assumed that the population the patients were drawn from had a known joint prior probability distribution over all drug clearances and volumes possible. The time sensitivity of the drug infusion problem was modelled using a discount factor $\alpha = e^{-\beta L}$, which was also used to approximate a finite horizon of unknown (but long) length. The control space D contained infusion rates, and a system function f was used to generate a new state $\left(v, l, {U_d}^{(D)}(C)\right)$ from the current state (v, l, C) and action taken d, where ${U_d}^{(D)} = e^{-\frac{l}{v}L}C + \frac{d}{l}(1 - e^{-\frac{l}{v}L})$ [27]. The system function could be used to determine the concentration of drug in the system if the exact volume and clearance for the patient was known, making the problem a much simpler issue of choosing an optimal control policy. A one-stage cost function g imposed penalties for deviations from a target concentration of drug in the system (unspecified). The optimal policy (which minimizes total cost) for the concentration of drug in the patient's system, can be represented using the following dynamic programming recursion [27]:

$$HW(c, p) = \min_{d}\{\int g(v, l, c(v, l), d)dp(v, l) + \alpha E_Z W(c', p')\} \tag{A.1}$$

where z represents an observation (a measurement of the patient's drug concentration in the bloodstream), and the conditional expectation regarding this observation given c,p, and d is denoted by E_Z. It would be ideal if this type of information was available for inclusion in feature selection for predictive pain management, as the concentration of opiates in the patient's blood would likely allow very accurate prediction of the likelihood of a patient being able to experience pain. While this may not be feasible because of the invasive measures necessary to obtain this information, predicting blood concentration indirectly (using the probable clearance rate in combination with the patient's physical characteristics) could be a valuable addition to models for predictive pain management.

This model is both predictive and uses time-varying features to monitor the need for analgesia administration. As such, of all of the work previously described, this model is most closely related to the goals of our proposed application of gLOP. However, using pharmacokinetic and pharmacodynamic measures is not practical in the context of patient monitoring in post operative recovery. Patients are monitored more passively in a recovery setting compared to surgery, so this type of information would not be readily available. Additionally, expert opinion would likely be needed for the parameterization of this type of system in practice, and we aim to build a model that uses only commonly measured physiological inputs and does not require expert knowledge for parameterization. However, even with these limitations regarding the applicability of this work to the problem of predictive pain management, we may able to base aspects of our future model for controlling analgesia administration on some of the modelling structures and assumptions introduced in this study, albeit with different inputs and different outputs.

A.4 Model Features and Feature Selection

Based on the extensive literature review performed, various physiological indicators have been identified as useful in assessing the level of pain a patient is in, especially for special patient populations. These would ideally be studied more vigorously using feature selection techniques to determine which indicators are most useful for predicting analgesia consumption [9]. Additionally, monitoring vital signs and other physiological indicators is common practice in clinical settings, depending on the health status or disease severity of a patient, making these quantitative features feasible to collect or obtain from existing data sources. Several features were identified as being useful in the context of prediction of future analgesia use in particular: these include gender, tobacco use, surgery characteristics such as type of analgesia used, anaesthesiologist, surgeon, operation room location, start time, etc; and characteristics of PCA use if predicting PCA adjustment. Many of the

studies looked at did not focus on continuous physiological indicators such as those found in waveform data. For this reason, it would be idea to explore the suitability of measures such as blood pressure, O_2 saturation, and ECG for PPM.

Significant study has been put into the subject of assessing the amount of pain a patient is currently experiencing using behavioural indicators. While these assessments may be subjective, including an explicit estimation of the amount of pain a patient is in would be valuable for training for a model in this context. Like some physiological measurements, when pain is being monitored it is common for clinical practitioners to ask patients how much pain they are experiencing or record a judgement of pain based on patient indicators; for this reason, it would be relatively easy to collect or obtain this information from patient records. Specifically, behavioural pain scores or a numeric rating score of pain at rest and while coughing would be interesting features to explore for PPM, even if some margin of error is included because of the subjective nature of this assessment.

Finally, while facial expression analysis using machine vision techniques can provide a rich source of information for determining whether a patient is currently experiencing pain, capturing photos and videos of patients in a clinical setting poses ethical concerns that might make such data very difficult to record or obtain. Additionally, the manual annotation and processing of this data can require a prohibitive amount of time and effort, and would limit the ability of a model to predict analgesia requirements in real-time. For this reason, features derived from facial expression recorded via visual media are not recommended to be included in feature selection for predictive pain management at this time. Many behavioural pain inventories include items detailing facial expression, so it may still be possible to capture this material, albeit through indirect sources. Measures related to direct pain assessment using behavioural inventories or facial expression detection are not commonly used in models for predicting future analgesia requirements. However, as the usefulness of such features in this context has not yet been determined, it would be prudent to explore their suitability for PPM, depending on the availability and ease of obtaining this information in a time-varying format.

A.5 Conclusions

Based on the literature reviewed, it is evident that significant work remains to be done towards building accurate temporal models of analgesia consumption for predictive pain management in a post-operative recovery setting. A summary of this work is provided in Table A.2 and Table A.3 Previous work on inferring current pain status in the medical community has focused largely on manual identification of pain using subjective behavioural

indicators or patient self-report. This is a methodological limitation because of deficits in clinical skills related to pain management evident in studies evaluating nurses and clinical practitioners. Using such subjective measures may also hinder proper pain management in cases where patients may not be able to communicate with physicians to indicate that they are in pain and require the administration of analgesia. In contrast, the machine learning community has focused on using facial expression monitoring to infer current pain status. This information would likely not be available in a post-operative recovery setting due to ethical considerations of capturing photos and videos of patients. It is also unclear as to whether these methods would be informative for predicting analgesia requirements as they only focus on current identification of pain. In general these models not predictive and do not use time-varying inputs.

More related to the future application of gLOP, studies predicting treatment demand for pain management were predictive but mostly did not use time-varying inputs [51, 52]. These studies were largely data-driven in nature, without real theoretical justification for particular statistical methods used in relation to the nature of the problem being studied. The final study described was most related to predictive pain management, as it was both predictive and used fine-grained time-varying inputs. However, this model used pharmacokinetic and pharmacodynamic features, and these will likely not be available to us for use in a PPM context. Because our future goal is to study analgesia consumption in a post operative setting instead of during surgery, patients will be monitored less closely, using less invasive techniques.

In general, while none of the work previously describes addresses the exact goal of predicting pain and thus analgesia consumption at a future point in time, and do not provide a model suitable for controlling analgesia administration specifically, the results of the related studies performed will guide our future work in developing a model for this application, in both model structure and the inputs used for feature selection. Our algorithm gLOP provides an avenue for selecting relevant features for the purpose of predictive pain management, but the previous work in the field management will guide our choice of features to use as input for this.

Table A.2: Summary of previous work in pain prediction and assessment

Author	Goal(s)	Methods	Significant Features
Tighe 2011	Predict whether patients will need FNB and other analgesia after ACL repair (Future)	Boosted logistic regression Bayes net Multilayer perceptron SVM ADTrees	Gender Tobacco use Types of analgesia used during surgery
Tighe 2012	Predict whether patients will need Acute Pain Service consultation (Future)	Bayes net Nave bayes Radial basis frequency network Voted perceptron KNN Decision trees Propositional rule learner PART decision list ZeroR J48 ADTree Random forest	Anaesthesiologist Surgeon Primary and secondary procedural terminology codes Start time Location of operating room
Hu, 1996	Model and control concentration of analgesia in bloodstream during surgery (Not feature selection) (Present)	Dynamic programming Infinite horizon POMDP with biological compartment model	Compartment volume Pharmacokinetic parameters Drug clearance rates Drug infusion rate
Hu, 2012	1. Categorize post-operative analgesia into three symbolic categories 2. Predict whether patients will require PCA settings adjustment in 48h after surgery (Future)	1. C4.5 decision tree C4.5 Bagging C4.5 Adaboost ensemble ANN SVM Random forest Rotational forest Nave bayes 2. Decision trees with bagging and boosting Oversampling and undersampling inbalanced class	1. PCA dose (var h) Time diff mean (var h) Time diff variance (19h) 2. Cont PCA dose (24h) Time diff mean (var h) Time diff variance (var h) PCA mode setting (var h) Operation time Patient weight Systolic BP Pulse

Table A.3: Summary of previous work in pain prediction and assessment (continued)

Author	Goal(s)	Methods	Significant Features
Becouze, 2007	Detect when patient is in pain via facial wrinkling (Not feature selection) (Present)	1. Detect head position 2. Put boundary around face in image 3. Segment face 4. Evaluate grimacing 5. Compute agitation based on facial expression, hr, bp, and other factors related to agitation	Grimacing Presence of wrinkles
Gholami, 2010	Classify image of neonate as being in pain/ not in pain (Present)	Sparse kernel (relevance vector) machine	Pain: Extract blood from infant's heel by puncturing Not pain: Puff of air Crib transport Friction (rub heel with alcohol)
Valstar, 2007	Identify which of 4 temporal phases face is in: Onset (face starts to change) Apex (expression at its peak) Offset (muscles relaxing) Neutral (Present)	HMM in combination with SVM to capture temporal dynamics	20 key facial points found using particle filtering

Appendix B

Preliminary Analysis and MIMIC-II Challenges

In preliminary work for this thesis, significant effort was made to make use of the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-II) project [36, 19], in order to provide a demonstration of our method with data in the context of our proposed application, predictive pain management. A particularly attractive feature of this data set is the temporal matching of patient clinical records to physiological waveforms collected during the patient's stay at the hospital. Based on the extensive literature review performed in Appendix A, we attempted to choose features consistent with pain as described in previous work (such as blood pressure and pulse [28]), as well as physiological responses that change specifically in response to the administration of opioids, such as respiration [55, 45, 13]. However, several aspects of the clinical records made this task particularly challenging. The first of these was the inconsistency in encoding of various events by different hospitals and practitioners. A specific example of this is events for wound dressing; this was alternately encoded in the chart records as "drsg", "dressing", "dsg", and "drsng". Similarly, inspired pressure was alternately encoded as "p.high", "p.hi", "press high", "p-hi", "p hi", "hi p", "p-high", "p high", "phigh", "high p", "pressure hi", "pressure high", "hi pressure", "high pressure", "p-high", "p-hi", and "pres. hi". While the issue of alternate encodings for the same event was fairly easily remedied by parsing the data using regular expressions, a certain amount of interpretation was required as to what event a code referred to. This was challenging, especially given our lack of access to a physician. Additionally, misspellings and similar issues meant that a percentage of codes had to be parsed manually, which was a time-consuming endeavour.

Another challenging aspect of the data set was understanding the meaning of the various

time codes for the purposes of establishing the specific timing of analgesia administration events. There were two time codes; one of these was the actual time of the event, and the other was the time that the event was entered into the chart. Ideally we had hoped for data in which the patient was self-administrating opioids based on pain experienced following surgery. However, even though many analgesia administration events were labeled as "PCA" events, the timing of these events was extremely regular (i.e. every hour on the hour for some patients), which lead us to believe that the PCA machines were being used to automatically administer analgesia to the patient at arbitrary doses as many of the patients would have been unconscious during their stay in the ICU. This conclusion was reached in consultation with medical experts at the Meaningful Use of Complex Medical Data (MUCMD) conference in 2013 [55, 45, 13].

Because the physiological measurements recorded in the chart records were very sparse, we opted to instead select patients in the data set with the same administration method and type of analgesia with matched waveform records and clinical records and extract windows of waveforms around the analgesia events matching our criteria. We chose fentanyl as the preferred analgesia for this analysis, because it has a faster onset and shorter duration than many of the alternative opioids such as morphine [8, 41]. For patients matching this criteria, sections of cardiac waveforms were extracted around the event (we called these "windows"). The waveforms forming the "negative" observations were those directly preceding an administration of analgesia. We chose this window to span from 8 minutes to 6 minutes prior to an administration of fentanyl; we hypothesized that this was in the middle of the range where physiological signals may have to changed to reflect increased pain [8, 41], as the previous dose of fentanyl (depending on dosage spacing) would have worn off in all of the events we observed. Similarly, because the onset of fentanyl is from 5-10 minutes after administration [8, 41], we chose our "positive" events (events post anaglesia administration) to span from 6 minutes to 8 minutes following an administration of fentanyl. For both positive and negative events, these windows were separated into consecutive chunks of waveforms 15 seconds in duration; this was to increase the effective number of events available for analysis, given that generally the number of fentanyl administrations observed per patient was from approximately 10 events to 70 events (with a few outliers). For each of these small waveform chunks, we extracted both Gaussian and Morlet wavelet coefficients (only the real component of the Morlet coefficients was used for analysis). These became the inputs for the model for this data.

Unfortunately, very few (only 4) patients in the data set matched the criteria of having matched cardiac waveforms and clinical records and having fentanyl administration events delivered via the PCA machine. Additional exclusion criteria included a previous history chronic pain, substance abuse, or previous chronic use of opioids. Other patients may have

matched the criteria, but in many cases the type of analgesia administered during the PCA events was not specified. The small number of patients matching our criteria was also problematic for analysis due to the heterogeneity of patients in our small sample size. Ideally, we would have only patients without gastric and renal issues, or patients with the same type of gastric or renal issues in our sample population, given that gastric and renal problems affect pharmacokinetics [8, 41], but there were not enough patients matching our criteria to separate patients based on this type of concern. Similarly, cardiac abnormalities may also affect pharmacokinetics. This type of variability in the patient population could have potentially had a negative effect on the quality of the results found via this analysis.

Initial experiments with simple individual Lassos for each patient yielded extremely poor results; this may have been caused by several factors. It is possible that of our choice of window timing or the length of the windows was not appropriate for this analysis and did not yield enough signal for relevant features to be selected by the Lasso paradigm. Additionally, it is possible that the cardiac waveform used (selected because it was the only waveform common to all of the patients matching our criteria) was not a strong enough predictor of the anaglesia events, and thus the wavelet features derived from the cardiac waveforms were also not strong predictors of our target. The cardiac waveforms used may have also included a significant amount of noise, resulting in a low signal-to-noise ratio which would impede feature selection. We examined the covariance matrix of the wavelet features, and also found that some wavelet features were extremely highly correlated. This may have resulted in relevant features being excluded from the model because of cancellation effects; correlations are problematic for the Lasso in this context, particularly when the "true" coefficients for correlated features have opposite signs and roughly equal magnitudes. Another potential reason for the poor results observed could have been the ratio of p to N; we had a huge number of features and a relatively small number of observations per patient, which would also have been a challenging context for performing feature selection.

While we were not able to perform a satisfactory analysis using this data set, we were able to base the synthetic data sets used to test gLOP off of the data characteristics observed from this analysis. Specifically, we chose a relatively small number of observations per patient (i.e. setting), and varied the number of features to explore the effect of these on the modelling capabilities of both gLOP and the dirty model, while holding the number of observations constant. We observed that the wavelet features derived from the patient waveforms were correlated, but chose to use non-correlated synthetic data in order to obtain clearer results without the issue of correlation for our initial analysis of gLOP. This is an avenue of future work that we intend to explore; having a model with the ability to deal with correlated features will be very valuable for predictive pain management.

References

[1] Andreas Alfons. *cvTools: Cross-validation tools for regression models*, 2012. R package version 0.3.2. Available online at http://CRAN.R-project.org/package=cvTools.

[2] Veldina Altenhein Howard and Frances W. Thurber. The interpretation of infant pain: Physiological and behavioral indicators used by NICU nurses. *Journal of Pediatric Nursing*, 13(3):164–174, June 1998.

[3] M. Arif-Rahu and M.J. Grap. Facial expression and pain in the critically ill non-communicative patient: State of science review. *Intensive and critical care nursing*, 26(6):343–352, 2010.

[4] Pierrick Becouze, Christopher E Hann, J Geoffrey Chase, and Geoffrey M Shaw. Measuring facial grimacing for quantifying patient agitation in critical care. *Computer methods and programs in biomedicine*, 87(2):138–147, 2007.

[5] Bert Bonroy, Greet Leysens, Dragana Miljkovic, Pieter Schiepers, Eric Triau, Maartje Wils, Daniel Berckmans, Patrick Colleman, LD Maesschalck, Stijn Quanten, et al. Image acquisition system to monitor discomfort in demented elderly patients. In *3rd Eur. Conf. Use Mod. Inf. Commun. Technol., Ghent, Belgium*, 2008.

[6] Dhruba Borthakur. The hadoop distributed file system: Architecture and design. *Hadoop Project Website*, 11:21, 2007.

[7] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[8] Feng Chang. Opioid pharmacokinetics and patient sample selection. Personal communication, 2013.

[9] Chi Wai Cheung, Chee Lun A Ying, Libby HY Lee, Suk Fung Tsang, Siu Lun Tsui, and Michael G Irwin. An audit of postoperative intravenous patient-controlled analgesia

with morphine: Evolution over the last decade. *European Journal of Pain*, 13(5):464–471, 2009.

[10] Yuan-Yi Chia, Lok-Hi Chow, Chun-Chieh Hung, Kang Liu, Luo-Ping Ger, and Pei-Ning Wang. Gender and pain upon movement are associated with the requirements for postoperative patient-controlled iv analgesia: a prospective survey of 2,298 chinese patients. *Canadian journal of anaesthesia = Journal canadien d'anesthsie*, 49(3):249–255, March 2002. PMID: 11861342.

[11] S J Dolin, J N Cashman, and J M Bland. Effectiveness of acute postoperative pain management: I. evidence from published data. *British journal of anaesthesia*, 89(3):409–423, September 2002. PMID: 12402719.

[12] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[13] Jesse Ehrenfeld. Analgesia administration in the icu and use of patient data. Personal communication, 2013.

[14] Margaret P. Ekstein and Avi A. Weinbroum. Immediate postoperative pain in orthopedic patients is more intense and requires more analgesia than in post-laparotomy patients. *Pain Medicine*, 12(2):308313, 2011.

[15] Michael J Fine, Daniel E Singer, Barbara H Hanusa, Judith R Lave, and Wishwa N Kapoor. Validation of a pneumonia prognostic index using the medisgroups comparative hospital database. *The American journal of medicine*, 94(2):153–159, 1993.

[16] LLdiko E Frank and Jerome H Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

[17] Cline Gelinas, Lise Fillion, Kathleen A. Puntillo, Chantal Viens, and Martine Fortier. Validation of the critical-care pain observation tool in adult patients. *American Journal of Critical Care*, 15(4):420–427, July 2006.

[18] B. Gholami, W.M. Haddad, and A.R. Tannenbaum. Relevance vector machine learning for neonate pain intensity assessment using digital imaging. *Biomedical Engineering, IEEE Transactions on*, 57(6):1457–1466, 2010.

[19] Glass L Hausdorff JM Ivanov PCh Mark RG Mietus JE Moody GB Peng C-K Stanley HE Goldberger AL, Amaral LAN. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, June 2000.

[20] Jeffrey A Grass. Patient-controlled analgesia. *Anesthesia & Analgesia*, 101(5S Suppl):S44–S61, 2005.

[21] Trevor Hastie and Brad Efron. *lars: Least Angle Regression, Lasso and Forward Stagewise*, 2013. R package version 1.2.

[22] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, chapter Linear Methods for Regression. Volume 2 of [23], 2009.

[23] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[24] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, chapter Unsupervised Learning. Volume 2 of [23], 2009.

[25] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[26] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[27] C. Hu, W.S. Lovejoy, and S.L. Shafer. Comparison of some suboptimal control policies in medical drug therapy. *Operations Research*, 44(5):696–709, 1996.

[28] Yuh-Jyh Hu, Tien-Hsiung Ku, Rong-Hong Jan, Kuochen Wang, Yu-Chee Tseng, and Shu-Fen Yang. Decision tree-based learning to predict patient controlled analgesia consumption and readjustment. *BMC Medical Informatics and Decision Making*, 12(1):131, November 2012.

[29] Hui Yun Vivian Ip, Amir Abrishami, Philip W. H. Peng, Jean Wong, and Frances Chung. Predictors of postoperative pain and analgesic consumption. *Anesthesiology*, 111(3):657–677, September 2009.

[30] Ali Jalali. L1linf_lasso.r, 2010. Available online at http://ali-jalali.com/index_files/L1Linf_LASSO.r.

[31] Ali Jalali, Pradeep D Ravikumar, Sujay Sanghavi, and Chao Ruan. A dirty model for multi-task learning. In *NIPS*, volume 3, page 7, 2010.

[32] Joel Katz and Zeev Seltzer. Transition from acute to chronic postsurgical pain: risk factors and protective factors. *Expert review of neurotherapeutics*, 9(5):723–744, 2009.

[33] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[34] Abhishek Kumar and Hal Daume III. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417*, 2012.

[35] Miriam Kunz, Siegfried Scharmann, Uli Hemmeter, Karsten Schepelmann, and Stefan Lautenbacher. The facial expression of pain in patients with dementia. *Pain*, 133(1):221–228, 2007.

[36] A.T. Reisner G. Clifford L. Lehman G.B. Moody T. Heldt T.H. Kyaw B.E. Moody-R.G. Mark M. Saeed, M. Villarroel. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): A public-access icu database. *Critical Care Medicine*, 39(5):952–652, May 2011.

[37] D Motamedvaziri, V Saligrama, and D Castanon. A combined approach to multi-label multi-task learning. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 616–619. IEEE, 2012.

[38] Sahand Negahban and Martin J Wainwright. Joint support recovery under high-dimensional scaling: Benefits and perils of l1,-regularization. *Advances in Neural Information Processing Systems*, 21:1161–1168, 2008.

[39] Guillaume Obozinski, Martin J Wainwright, Michael I Jordan, et al. Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, 39(1):1–47, 2011.

[40] Peter H. M.D. Pan, Robert Ph.D. Coghill, Timothy T. Ph.D. Houle, Melvin H. M.D. Seid, W Michael M.D. Lindel, R Lamar M.D. Parker, Scott A. M.D. Washburn, Lynne B.S.N. Harris, and James C. M.D. Eisenach. Multifactorial preoperative predictors for postcesarean section pain and analgesic requirement. *Anesthesiology*, 104(3):417–425, 2006.

[41] Tejal Patel. Opioid pharmacokinetics and patient sample selection. Personal communication, 2013.

[42] J.F. Payen, O. Bru, J.L. Bosson, A. Lagrasta, E. Novel, I. Deschaux, P. Lavagne, and C. Jacquot. Assessing pain in critically ill sedated patients by using a behavioral pain scale. *Critical care medicine*, 29(12):2258–2263, 2001.

[43] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. Available online at http://www.R-project.org/.

[44] Manon Ranger, C Celeste Johnston, and KJS Anand. Current controversies regarding pain assessment in neonates. In *Seminars in perinatology*, volume 31, pages 283–288. Elsevier, 2007.

[45] Warren Sandberg. Analgesia administration in the icu and use of patient data. Personal communication, 2013.

[46] Nicolas Simon, Saik Urien, Bruno Riou, Philippe Lechat, Frédéric Aubrun, et al. Intravenous morphine titration in immediate postoperative pain management: Population kinetic–pharmacodynamic and logistic regression analysis. *Pain*, 144(1):139–146, 2009.

[47] Ulrike M Stamer and Frank StÜber. The pharmacogenetics of analgesia. 2007.

[48] Mervyn Stone and Rodney J Brooks. Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 237–269, 1990.

[49] Billy Michael Thorne and J Martin Giesen. *Statistics for the behavioral sciences*. Mayfield Publishing Company, 1997.

[50] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[51] Patrick Tighe, Sarah Laduzenski, David Edwards, Neal Ellis, Andre P. Boezaart, and Haldun Aygtug. Use of machine learning theory to predict the need for femoral nerve block following ACL repair. *Pain Medicine*, 12(10):15661575, 2011.

[52] Patrick J. Tighe, Stephen D. Lucas, David A. Edwards, Andre P. Boezaart, Haldun Aytug, and Azra Bihorac. Use of machine-learning classifiers to predict requests for preoperative acute pain service consultation. *Pain Medicine*, 13(10):13471357, 2012.

[53] Michel F. Valstar and Maja Pantic. Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In *Proceedings of the 2007 IEEE international conference on Human-computer interaction*, HCI'07, page 118127, Berlin, Heidelberg, 2007. Springer-Verlag.

[54] Judy Watt-Watson, Bonnie Stevens, Paul Garfinkel, David Streiner, and Ruth Gallop. Relationship between nurses pain knowledge and pain management outcomes for their postoperative cardiac patients. *Journal of Advanced Nursing*, 36(4):535545, 2001.

[55] Randall Wetzel. Analgesia administration in the icu and use of patient data. Personal communication, 2013.

[56] Hilde Wøien and Ida Torunn Bjørk. Intensive care pain treatment and sedation: Nurses experiences of the conflict between clinical judgement and standardised care: An explorative study. *Intensive and Critical Care Nursing*, 2012.

[57] SJ Wright and J Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

[58] Christopher L Wu and Srinivasa N Raja. Treatment of acute postoperative pain. *The Lancet*, 377(9784):2215–2225, 2011.

[59] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[60] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pages 10–10, 2010.

[61] Peng Zhao, Guilherme Rocha, and Bin Yu. Grouped and hierarchical model selection through composite absolute penalties. *Department of Statistics, UC Berkeley, Tech. Rep*, 703, 2006.

[62] Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, pages 3468–3497, 2009.